



Open Source Used In AppDynamics_OTIS_Pipeline 23.8.2

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1767549848

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this [form](#).

In your requests please include the following reference number 78EE117C99-1767549848

Contents

1.1 libjpeg 6b

1.1.1 Notifications

1.1.2 Available under license

1.2 javax-inject 1

1.2.1 Available under license

1.3 commons-math 2.1

1.3.1 Available under license

1.4 commons-collections 3.2.2

1.4.1 Available under license

1.5 jeromq 0.3.5

1.5.1 Available under license

1.6 annotations 13.0

1.6.1 Available under license

1.7 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava

1.8 failureaccess 1.0.1

1.9 j2objc-annotations 1.3

1.9.1 Available under license

1.10 commons-digester 2.1

1.10.1 Available under license

1.11 lcms 2.9

1.11.1 Available under license

1.12 bean-validation-api 2.0.1.Final

1.12.1 Available under license

1.13 giflib 5.2.1

1.13.1 Available under license

1.14 event-stream 1.0.1

- 1.14.1 Available under license
- 1.15 scala 2.12.10**
 - 1.15.1 Available under license
- 1.16 error_prone_annotations 2.3.4**
 - 1.16.1 Available under license
- 1.17 jctools-core 3.1.0**
 - 1.17.1 Available under license
- 1.18 okio 2.8.0**
 - 1.18.1 Available under license
- 1.19 xerces-j 2.12.2**
 - 1.19.1 Available under license
- 1.20 httpcomponents-client 4.5.13**
 - 1.20.1 Available under license
- 1.21 guava 30.1-jre**
 - 1.21.1 Available under license
- 1.22 httpcomponents-core 4.4.13**
 - 1.22.1 Available under license
- 1.23 httpcore-nio 4.4.13**
 - 1.23.1 Available under license
- 1.24 httpclient-cache 4.5.13**
 - 1.24.1 Available under license
- 1.25 commons-logging 1.2**
 - 1.25.1 Available under license
- 1.26 commons-validator 1.7**
 - 1.26.1 Available under license
- 1.27 commons-io 2.8.0**
 - 1.27.1 Available under license
- 1.28 commons-codec 1.14**
 - 1.28.1 Available under license
- 1.29 javassist 3.27.0-GA**
 - 1.29.1 Available under license
- 1.30 zt 1.14**
 - 1.30.1 Available under license
- 1.31 httpcomponents-mime 4.5.12**
 - 1.31.1 Available under license
- 1.32 commons-codec 1.15**
 - 1.32.1 Available under license
- 1.33 joda-time 2.10.2**
 - 1.33.1 Available under license

- 1.34 snappy-java 1.1.8.4**
 - 1.34.1 Available under license
- 1.35 lz4 1.9.3**
 - 1.35.1 Available under license
- 1.36 asm 9.1**
 - 1.36.1 Available under license
- 1.37 lombok 1.18.20**
 - 1.37.1 Available under license
- 1.38 lombok-utils 1.18.12**
 - 1.38.1 Available under license
- 1.39 commons-lang3 3.12.0**
 - 1.39.1 Available under license
- 1.40 commons-compress 1.21**
 - 1.40.1 Available under license
- 1.41 free-type 2.10.4+dfsg-1**
 - 1.41.1 Available under license
- 1.42 namespace 1.4.01**
 - 1.42.1 Available under license
- 1.43 alpine-keys 2.4-r1**
 - 1.43.1 Available under license
- 1.44 jackson 2.15.0**
 - 1.44.1 Available under license
- 1.45 log4j-api 2.17.1**
 - 1.45.1 Available under license
- 1.46 apache-log4j 2.17.1**
 - 1.46.1 Available under license
- 1.47 log4j-jcl 2.17.1**
 - 1.47.1 Available under license
- 1.48 dom 1.0**
 - 1.48.1 Available under license
- 1.49 dagger 2.4**
 - 1.49.1 Available under license
- 1.50 paranamer 2.5.1**
 - 1.50.1 Available under license
- 1.51 avro 1.11.0**
 - 1.51.1 Available under license
- 1.52 annotations 4.1.1.4**
 - 1.52.1 Available under license
- 1.53 animal-sniffer-annotation 1.19**

- 1.53.1 Available under license
- 1.54 error_prone_annotations 2.3.3**
 - 1.54.1 Available under license
- 1.55 picocontainer 2.15**
 - 1.55.1 Available under license
- 1.56 handy-uri-templates 2.1.8**
 - 1.56.1 Available under license
- 1.57 javax-annotation-api 1.3.2**
 - 1.57.1 Available under license
- 1.58 error_prone_annotations 2.10.0**
 - 1.58.1 Available under license
- 1.59 gson 2.9.0**
 - 1.59.1 Available under license
- 1.60 guava 31.1-jre**
 - 1.60.1 Available under license
- 1.61 javapoet 1.13.0**
 - 1.61.1 Available under license
- 1.62 guava 31.0.1-android**
 - 1.62.1 Available under license
- 1.63 animal-sniffer-annotation 1.21**
 - 1.63.1 Available under license
- 1.64 ion-java 1.0.2**
 - 1.64.1 Available under license
- 1.65 gson 2.8.9**
 - 1.65.1 Available under license
- 1.66 perfmark-api 0.25.0**
 - 1.66.1 Available under license
- 1.67 zstd-jni 1.5.2-1**
 - 1.67.1 Available under license
- 1.68 okhttp 4.10.0**
 - 1.68.1 Available under license
- 1.69 reactive-streams 1.0.4**
 - 1.69.1 Available under license
- 1.70 url-connection-client 2.17.122**
 - 1.70.1 Available under license
- 1.71 mbknor-jackson-jsonschema_2.12 1.0.39**
 - 1.71.1 Available under license
- 1.72 kotlin-scripting-compiler-impl-embeddable 1.7.10**
 - 1.72.1 Available under license

1.73 zstd 1.5.2

1.73.1 Available under license

1.74 protobuf-java 3.19.6

1.74.1 Available under license

1.75 kotlin 1.6.21

1.75.1 Available under license

1.76 libpng 1.6.38

1.76.1 Available under license

1.77 zlib 1.2.13-r1

1.77.1 Available under license

1.78 jctools-core 4.0.1

1.78.1 Available under license

1.79 scala 2.12.10.v20190904-150159-VFINAL-61701c2

1.79.1 Available under license

1.80 slf4j 1.6.6

1.80.1 Available under license

1.81 protobuf-java-util 3.21.10

1.81.1 Available under license

1.82 jackson 2.14.2

1.82.1 Available under license

1.83 jackson-annotations 2.14.2

1.83.1 Available under license

1.84 jackson-databind 2.14.2

1.84.1 Available under license

1.85 wire-compiler 3.7.1

1.85.1 Available under license

1.86 glue 2.17.122

1.86.1 Available under license

1.87 schema-registry-serde 1.1.13

1.87.1 Available under license

1.88 schema-registry-build-tools 1.1.13

1.88.1 Available under license

1.89 sts 2.17.122

1.89.1 Available under license

1.90 arns 2.17.122

1.90.1 Available under license

1.91 kotlin-scripting-compiler-embeddable 1.7.10

1.91.1 Available under license

1.92 wire-kotlin-generator 3.7.1

- 1.92.1 Available under license
- 1.93 schema-registry-common 1.1.13**
 - 1.93.1 Available under license
- 1.94 jackson-dataformat-cbor 2.14.2**
 - 1.94.1 Available under license
- 1.95 kotlin 1.8.10-release-430**
 - 1.95.1 Available under license
- 1.96 wire-profiles 3.7.1**
 - 1.96.1 Available under license
- 1.97 wire-grpc-server-generator 3.7.1**
 - 1.97.1 Available under license
- 1.98 wire-swift-generator 3.7.1**
 - 1.98.1 Available under license
- 1.99 wire-java-generator 3.7.1**
 - 1.99.1 Available under license
- 1.100 amazon-kinesis-client 2.4.5**
 - 1.100.1 Available under license
- 1.101 netty-nio-client 2.19.2**
 - 1.101.1 Available under license
- 1.102 commons-fileupload 1.5**
 - 1.102.1 Available under license
- 1.103 tre 1.2.3-r4**
 - 1.103.1 Available under license
- 1.104 pgv-java-stub 0.6.1**
 - 1.104.1 Available under license
- 1.105 apache-log4j 2.20.0**
 - 1.105.1 Available under license
- 1.106 log4j-api 2.20.0**
 - 1.106.1 Available under license
- 1.107 log4j-slf4j2-impl 2.20.0**
 - 1.107.1 Available under license
- 1.108 error_prone_annotations 2.18.0**
 - 1.108.1 Available under license
- 1.109 opentelemetry-proto 0.11.0**
 - 1.109.1 Available under license
- 1.110 json-java 20201115**
 - 1.110.1 Available under license
- 1.111 everit-json-schema 1.12.2**
 - 1.111.1 Available under license

- 1.112 classgraph 4.8.120**
 - 1.112.1 Available under license
- 1.113 jimfs 1.1**
 - 1.113.1 Available under license
- 1.114 protobuf-java 3.22.2**
 - 1.114.1 Available under license
- 1.115 pgv-java-stub 0.9.1**
 - 1.115.1 Available under license
- 1.116 annotations 2.19.2**
 - 1.116.1 Available under license
- 1.117 protocol-core 2.19.2**
 - 1.117.1 Available under license
- 1.118 netty-transport-classes-epoll 4.1.90.Final**
 - 1.118.1 Available under license
- 1.119 third-party-jackson-core 2.19.2**
 - 1.119.1 Available under license
- 1.120 json-utils 2.19.2**
 - 1.120.1 Available under license
- 1.121 cloudwatch 2.19.2**
 - 1.121.1 Available under license
- 1.122 client-spi 2.19.2**
 - 1.122.1 Available under license
- 1.123 utils 2.19.2**
 - 1.123.1 Available under license
- 1.124 aws-cbor-protocol 2.19.2**
 - 1.124.1 Available under license
- 1.125 sdk-core 2.19.2**
 - 1.125.1 Available under license
- 1.126 netty-codec-socks 4.1.90.Final**
 - 1.126.1 Available under license
- 1.127 netty-resolver 4.1.90.Final**
 - 1.127.1 Available under license
- 1.128 netty-handler-proxy 4.1.90.Final**
 - 1.128.1 Available under license
- 1.129 netty-handler 4.1.90.Final**
 - 1.129.1 Available under license
- 1.130 proto-google-common-protos 2.11.0**
 - 1.130.1 Available under license
- 1.131 apache-client 2.19.2**

- 1.131.1 Available under license
- 1.132 profiles 2.19.2**
 - 1.132.1 Available under license
- 1.133 regions 2.19.2**
 - 1.133.1 Available under license
- 1.134 metrics-spi 2.19.2**
 - 1.134.1 Available under license
- 1.135 netty-codec-http 4.1.90.Final**
 - 1.135.1 Available under license
- 1.136 third-party-jackson-dataformat-cbor 2.19.2**
 - 1.136.1 Available under license
- 1.137 aws-json-protocol 2.19.2**
 - 1.137.1 Available under license
- 1.138 aws-core 2.19.2**
 - 1.138.1 Available under license
- 1.139 netty-transport 4.1.90.Final**
 - 1.139.1 Available under license
- 1.140 auth 2.19.2**
 - 1.140.1 Available under license
- 1.141 netty 4.1.90.Final**
 - 1.141.1 Available under license
- 1.142 endpoints-spi 2.19.2**
 - 1.142.1 Available under license
- 1.143 netty-codec 4.1.90.Final**
 - 1.143.1 Available under license
- 1.144 aws-query-protocol 2.19.2**
 - 1.144.1 Available under license
- 1.145 netty-transport-native-unix-common 4.1.90.Final**
 - 1.145.1 Available under license
- 1.146 kinesis 2.19.2**
 - 1.146.1 Available under license
- 1.147 dynamodb 2.19.2**
 - 1.147.1 Available under license
- 1.148 opentelemetry-context 1.24.0**
 - 1.148.1 Available under license
- 1.149 aws-java-sdk-core 1.12.430**
 - 1.149.1 Available under license
- 1.150 pgv-java-grpc 0.9.1**
 - 1.150.1 Available under license

1.151 kotlin-reflect 1.8.10-release-430

1.151.1 Available under license

1.152 jmespath-java 1.12.430

1.152.1 Available under license

1.153 aws-java-sdk-sts 1.12.430

1.153.1 Available under license

1.154 opentelemetry-api 1.24.0

1.154.1 Available under license

1.155 re2j 1.7

1.155.1 Available under license

1.156 kotlinpoet 1.7.2

1.156.1 Available under license

1.157 xorg-randr 21.1.8-r0

1.157.1 Available under license

1.158 bridgeutils 1.7.1-r2

1.158.1 Available under license

1.159 pax-utils 1.3.7-r1

1.159.1 Available under license

1.160 jre 17.0.7+7-LTS

1.160.1 Available under license

1.161 jsr305 3.0.2

1.161.1 Available under license

1.162 xml-apis 1.4.01

1.162.1 Available under license

1.163 servlet-api 3.0.1

1.163.1 Available under license

1.164 perfmark-api 0.17.0

1.164.1 Available under license

1.165 jackson-annotations 2.15.0

1.165.1 Available under license

1.166 jackson-dataformat-yaml 2.15.0

1.166.1 Available under license

1.167 jackson-databind 2.15.0

1.167.1 Available under license

1.168 disruptor 2.11.2.3

1.168.1 Available under license

1.169 openjdk 17.0.7

1.169.1 Available under license

1.170 lz4-java 1.8.0

- 1.170.1 Available under license
- 1.171 ca-certificates 20230506-r0**
 - 1.171.1 Available under license
- 1.172 apk-tools 2.14.0-r2**
 - 1.172.1 Available under license
- 1.173 alpine-baselayout 3.4.3-r1**
 - 1.173.1 Available under license
- 1.174 ca-certificates-bundle 20230506-r0**
 - 1.174.1 Available under license
- 1.175 libc-utils 0.7.2-r5**
 - 1.175.1 Available under license
- 1.176 jackson-annotations 2.15.1**
 - 1.176.1 Available under license
- 1.177 jackson-databind 2.15.1**
 - 1.177.1 Available under license
- 1.178 jackson 2.15.1**
 - 1.178.1 Available under license
- 1.179 jackson-dataformat-yaml 2.15.1**
 - 1.179.1 Available under license
- 1.180 alpine-baselayout-data 3.4.3-r1**
 - 1.180.1 Available under license
- 1.181 opentelemetry-sdk 1.24.0**
 - 1.181.1 Available under license
- 1.182 opentelemetry-semconv 1.24.0-alpha**
 - 1.182.1 Available under license
- 1.183 opentelemetry-sdk-extension-autoconfigure-spi 1.24.0**
 - 1.183.1 Available under license
- 1.184 scanelf 1.3.7-r1**
 - 1.184.1 Available under license
- 1.185 busybox 1.36.1**
 - 1.185.1 Available under license
- 1.186 musl 1.2.4-r1**
 - 1.186.1 Available under license
- 1.187 reload4j 2.20.0**
 - 1.187.1 Available under license
- 1.188 slf4j 2.0.7**
 - 1.188.1 Available under license
- 1.189 snake-yaml 2.6**
 - 1.189.1 Available under license

1.190 kotlin-reflect 1.8.10

1.190.1 Available under license

1.191 kotlin 1.8.10

1.191.1 Available under license

1.192 protobuf-java 3.22.5

1.192.1 Available under license

1.193 snake-yaml 2.0

1.193.1 Available under license

1.194 openjdk-jre 17.0.7u7

1.194.1 Available under license

1.195 kotlin-script-runtime 1.8.10

1.195.1 Available under license

1.196 kotlin-scripting-common 1.8.10

1.196.1 Available under license

1.197 kotlin-scripting-jvm 1.8.10

1.197.1 Available under license

1.198 busybox 1.36.1-r2

1.198.1 Available under license

1.199 pgv-java-grpc 0.6.1

1.199.1 Available under license

1.200 musl-utils 1.2.4-r1

1.200.1 Available under license

1.201 ssl-client 1.36.1-r2

1.201.1 Available under license

1.202 libssl3 3.1.2-r0

1.202.1 Available under license

1.203 libcrypto3 3.1.2-r0

1.203.1 Available under license

1.204 openssl 3.1.2-r0

1.204.1 Available under license

1.1 libjpeg 6b

1.1.1 Notifications :

This software is based in part on the work of the Independent JPEG Group.

1.1.2 Available under license :

The Independent JPEG Group's JPEG software

=====

README for release 6b of 27-Mar-1998

=====

This distribution contains the sixth public release of the Independent JPEG Group's free JPEG software. You are welcome to redistribute this software and to use it for any purpose, subject to the conditions under LEGAL ISSUES, below.

Serious users of this software (particularly those incorporating it into larger programs) should contact IJG at jpeg-info@uunet.uu.net to be added to our electronic mailing list. Mailing list members are notified of updates and have a chance to participate in technical discussions, etc.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group.

IJG is not affiliated with the official ISO JPEG standards committee.

DOCUMENTATION

ROADMAP

=====

This file contains the following sections:

OVERVIEW General description of JPEG and the IJG software.
LEGAL ISSUES Copyright, lack of warranty, terms of distribution.
REFERENCES Where to learn more about JPEG.
ARCHIVE LOCATIONS Where to find newer versions of this software.
RELATED SOFTWARE Other stuff you should get.
FILE FORMAT WARS Software *not* to get.
TO DO Plans for future IJG releases.

Other documentation files in the distribution are:

User documentation:

install.doc How to configure and install the IJG software.
usage.doc Usage instructions for cjpeg, djpeg, jpegtran,
 rdjpgcom, and wrjpgcom.
*.1 Unix-style man pages for programs (same info as usage.doc).
wizard.doc Advanced usage instructions for JPEG wizards only.
change.log Version-to-version change highlights.

Programmer and internal

documentation:

libjpeg.doc How to use the JPEG library in your own programs.
example.c Sample code for calling the JPEG library.
structure.doc Overview of the JPEG library's internal structure.
filelist.doc Road map of IJG files.
coderrules.doc Coding style rules --- please read if you contribute code.

Please read at least the files `install.doc` and `usage.doc`. Useful information can also be found in the JPEG FAQ (Frequently Asked Questions) article. See ARCHIVE LOCATIONS below to find out where to obtain the FAQ article.

If you want to understand how the JPEG code works, we suggest reading one or more of the REFERENCES, then looking at the documentation files (in roughly the order listed) before diving into the code.

OVERVIEW

=====

This package contains C software to implement JPEG image compression and decompression. JPEG (pronounced "jay-peg") is a standardized compression method for full-color and gray-scale images.

JPEG is intended for compressing "real-world" scenes; line drawings, cartoons and other non-realistic images are not its strong suit. JPEG is lossy, meaning that the output image is not exactly identical to the input image. Hence you must not use JPEG if you have to have identical output bits. However, on typical photographic images, very good compression levels can be obtained with no visible change, and remarkably high compression levels are possible if you can tolerate a low-quality image. For more details, see the references, or just experiment with various compression settings.

This software implements JPEG baseline, extended-sequential, and progressive compression processes. Provision is made for supporting all variants of these processes, although some uncommon parameter settings aren't implemented yet. For legal reasons, we are not distributing code for the arithmetic-coding variants of JPEG; see LEGAL ISSUES. We have made no provision for supporting the hierarchical or lossless processes defined in the standard.

We provide a set of library routines for reading and writing JPEG image files, plus two sample applications "cjpeg" and "djpeg", which use the library to perform conversion between JPEG and some other popular image file formats. The library is intended to be reused in other applications.

In order to support file conversion and viewing software, we have included considerable functionality beyond the bare JPEG coding/decoding capability; for example, the color quantization modules are not strictly part of JPEG decoding, but they are essential for output to colormapped file formats or colormapped displays. These extra functions can be compiled out of the library if not required for a particular application. We have also included "jpegtran", a utility for lossless transcoding between different JPEG processes, and "rdjpgcom" and "wrjpgcom", two simple applications for inserting and extracting textual comments

in JFIF files.

The emphasis in designing this software has been on achieving portability and flexibility, while also making it fast enough to be useful. In particular, the software is not intended to be read as a tutorial on JPEG. (See the REFERENCES section for introductory material.) Rather, it is intended to be reliable, portable, industrial-strength code. We do not claim to have achieved that goal in every aspect of the software, but we strive for it.

We welcome the use of this software as a component of commercial products. No royalty is required, but we do ask for an acknowledgement in product documentation, as described under LEGAL ISSUES.

LEGAL ISSUES

=====

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the

resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

REFERENCES

=====

We

highly recommend reading one or more of these references before trying to understand the innards of the JPEG software.

The best short technical introduction to the JPEG compression algorithm is Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (vol. 34 no. 4), pp. 30-44. (Adjacent articles in that issue discuss MPEG motion picture compression, applications of JPEG, and related topics.) If you don't have the CACM issue handy, a PostScript file containing a revised version of Wallace's article is available at <ftp://ftp.uu.net/graphics/jpeg/wallace.ps.gz>. The file (actually a preprint for an article that appeared in IEEE Trans. Consumer Electronics) omits the sample images that appeared in CACM, but it includes corrections and some added material. Note: the Wallace article is copyright ACM and IEEE, and it may not be used for commercial purposes.

A somewhat less technical, more leisurely introduction to JPEG can be found in "The Data Compression Book" by Mark Nelson and Jean-loup Gailly, published by M&T Books (New York), 2nd ed. 1996, ISBN 1-55851-434-1. This book provides good explanations and example C code for a multitude of compression methods including JPEG. It is an excellent source if you are comfortable reading C code but don't know much about data compression in general. The book's JPEG sample code is far from industrial-strength, but when you are ready to look at a full implementation, you've got one here...

The best full description of JPEG is the textbook "JPEG Still Image Data Compression Standard" by William B. Pennebaker and Joan L. Mitchell, published by Van Nostrand Reinhold, 1993, ISBN 0-442-01272-1. Price US\$59.95, 638 pp. The book includes the complete text of the ISO JPEG standards (DIS 10918-1 and draft DIS 10918-2). This is by far the most complete exposition of JPEG in existence, and we highly recommend it.

The

JPEG standard itself is not available electronically; you must order a paper copy through ISO or ITU. (Unless you feel a need to own a certified

official copy, we recommend buying the Pennebaker and Mitchell book instead; it's much cheaper and includes a great deal of useful explanatory material.) In the USA, copies of the standard may be ordered from ANSI Sales at (212) 642-4900, or from Global Engineering Documents at (800) 854-7179. (ANSI doesn't take credit card orders, but Global does.) It's not cheap: as of 1992, ANSI was charging \$95 for Part 1 and \$47 for Part 2, plus 7% shipping/handling. The standard is divided into two parts, Part 1 being the actual specification, while Part 2 covers compliance testing methods. Part 1 is titled "Digital Compression and Coding of Continuous-tone Still Images, Part 1: Requirements and guidelines" and has document numbers ISO/IEC IS 10918-1, ITU-T T.81. Part 2 is titled "Digital Compression and Coding of Continuous-tone Still Images, Part 2: Compliance testing" and has document numbers ISO/IEC IS 10918-2, ITU-T T.83.

Some extensions to the original JPEG standard are defined in JPEG Part 3, a newer ISO standard numbered ISO/IEC IS 10918-3 and ITU-T T.84. IJG currently does not support any Part 3 extensions.

The JPEG standard does not specify all details of an interchangeable file format. For the omitted details we follow the "JFIF" conventions, revision 1.02. A copy of the JFIF spec is available from:

Literature Department
C-Cube Microsystems, Inc.
1778 McCarthy Blvd.
Milpitas, CA 95035
phone (408) 944-6300, fax (408) 944-6314

A PostScript version of this document is available by FTP at <ftp://ftp.uu.net/graphics/jpeg/jfif.ps.gz>. There is also a plain text version at <ftp://ftp.uu.net/graphics/jpeg/jfif.txt.gz>, but it is missing the figures.

The TIFF 6.0 file format specification can be obtained by FTP from <ftp://ftp.sgi.com/graphics/tiff/TIFF6.ps.gz>. The JPEG incorporation scheme found in the TIFF 6.0 spec of 3-June-92 has a number of serious problems. IJG does not recommend use of the TIFF 6.0 design (TIFF Compression tag 6). Instead, we recommend the JPEG design proposed by TIFF Technical Note #2 (Compression tag 7). Copies of this Note can be obtained from <ftp.sgi.com> or from <ftp://ftp.uu.net/graphics/jpeg/>. It is expected that the next revision of the TIFF spec will replace the 6.0 JPEG design with the Note's design. Although IJG's own code does not support TIFF/JPEG, the free libtiff library uses our library to implement TIFF/JPEG per the Note. libtiff is available from <ftp://ftp.sgi.com/graphics/tiff/>.

ARCHIVE LOCATIONS

=====

The "official" archive site for this software is ftp.uu.net (Internet address 192.48.96.9). The most recent released version can always be found there in directory graphics/jpeg. This particular version will be archived as ftp://ftp.uu.net/graphics/jpeg/jpegsrsrc.v6b.tar.gz. If you don't have direct Internet access, UUNET's archives are also available via UUCP; contact help@uunet.uu.net for information on retrieving files that way.

Numerous Internet sites maintain copies of the UUNET files. However, only ftp.uu.net is guaranteed to have the latest official version.

You can also obtain this software in DOS-compatible "zip" archive format from the SimTel archives (ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/), or on CompuServe in the Graphics Support forum (GO CIS:GRAPHSUP), library 12 "JPEG Tools". Again, these versions may sometimes lag behind the ftp.uu.net release.

The JPEG FAQ (Frequently Asked Questions) article is a useful source of general information about JPEG. It is updated constantly and therefore is not included in this distribution. The FAQ is posted every two weeks to Usenet newsgroups comp.graphics.misc, news.answers, and other groups. It is available on the World Wide Web at <http://www.faqs.org/faqs/jpeg-faq/> and other news.answers archive sites, including the official news.answers archive at [rtfm.mit.edu: ftp://rtfm.mit.edu/pub/usenet/news.answers/jpeg-faq/](http://rtfm.mit.edu/pub/usenet/news.answers/jpeg-faq/). If you don't have Web or FTP access, send e-mail to mail-server@rtfm.mit.edu with body
send usenet/news.answers/jpeg-faq/part1
send usenet/news.answers/jpeg-faq/part2

RELATED SOFTWARE

=====

Numerous viewing and image manipulation programs now support JPEG. (Quite a few of them use this library to do so.) The JPEG FAQ described above lists some of the more popular free and shareware viewers, and tells where to obtain them on Internet.

If you are on a Unix machine, we highly recommend Jef Poskanzer's free PBPLUS software, which provides many useful operations on PPM-format image files. In particular, it can convert PPM images to and from a wide range of other formats, thus making cjpeg/djpeg considerably more useful. The latest version is distributed by the NetPBM group, and is available from numerous sites,

notably [ftp://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/](http://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/). Unfortunately PBPLUS/NETPBM is not nearly as portable as the IJG software is;

you are likely to have difficulty making it work on any non-Unix machine.

A different free JPEG implementation, written by the PVRG group at Stanford, is available from <ftp://havefun.stanford.edu/pub/jpeg/>. This program is designed for research and experimentation rather than production use; it is slower, harder to use, and less portable than the IJG code, but it is easier to read and modify. Also, the PVRG code supports lossless JPEG, which we do not. (On the other hand, it doesn't do progressive JPEG.)

FILE FORMAT WARS

=====

Some JPEG programs produce files that are not compatible with our library. The root of the problem is that the ISO JPEG committee failed to specify a concrete file format. Some vendors "filled in the blanks" on their own, creating proprietary formats that no one else could read.

(For example, none of the early commercial JPEG implementations for the Macintosh were able to exchange compressed files.)

The file format we have adopted is called JFIF (see REFERENCES). This format has been agreed to by a number of major commercial JPEG vendors, and it has become the de facto standard. JFIF is a minimal or "low end" representation. We recommend the use of TIFF/JPEG (TIFF revision 6.0 as modified by TIFF Technical Note #2) for "high end" applications that need to record a lot of additional data about an image. TIFF/JPEG is fairly new and not yet widely supported, unfortunately.

The upcoming JPEG Part 3 standard defines a file format called SPIFF. SPIFF is interoperable with JFIF, in the sense that most JFIF decoders should be able to read the most common variant of SPIFF. SPIFF has some technical advantages over JFIF, but its major claim to fame is simply that it is an official standard rather than an informal one. At this point it is unclear whether

SPIFF will supersede JFIF or whether JFIF will remain the de-facto standard. IJG intends to support SPIFF once the standard is frozen, but we have not decided whether it should become our default output format or not. (In any case, our decoder will remain capable of reading JFIF indefinitely.)

Various proprietary file formats incorporating JPEG compression also exist. We have little or no sympathy for the existence of these formats. Indeed, one of the original reasons for developing this free software was to help force convergence on common, open format standards for JPEG files. Don't use a proprietary file format!

TO DO

=====

The major thrust for v7 will probably be improvement of visual quality. The current method for scaling the quantization tables is known not to be very good at low Q values. We also intend to investigate block boundary smoothing, "poor man's variable quantization", and other means of improving quality-vs-file-size performance without sacrificing compatibility.

In future versions, we are considering supporting some of the upcoming JPEG Part 3 extensions --- principally, variable quantization and the SPIFF file format.

As always, speeding things up is of great interest.

Please send bug reports, offers of help, etc. to jpeg-info@uunet.uu.net.

1.2 javax-inject 1

1.2.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright (C) 2009 The JSR-330 Expert Group

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

1.3 commons-math 2.1

1.3.1 Available under license :

Apache Commons Math
 Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by
 The Apache Software Foundation (<http://www.apache.org/>).

=====

The LinearConstraint, LinearObjectiveFunction, LinearOptimizer, Relationship, SimplexSolver and SimplexTableau classes in package org.apache.commons.math.optimization.linear include software developed by Benjamin McCann (<http://www.benmccann.com>) and distributed with the following copyright: Copyright 2009 Google Inc.

=====

This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.
 The LevenbergMarquardtOptimizer class in package org.apache.commons.math.optimization.general includes software translated from the lmdcr, lmpar and qrsolv Fortran routines from the Minpack package
 Minpack Copyright
 Notice (1999) University of Chicago. All rights reserved

=====

The GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff includes software translated from the odex Fortran routine developed by E. Hairer and G. Wanner.
 Original source copyright:
 Copyright (c) 2004, Ernst Hairer

=====

The EigenDecompositionImpl class in package org.apache.commons.math.linear includes software translated from some LAPACK Fortran routines. Original source copyright:
 Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

The MersenneTwister class in package org.apache.commons.math.random includes software translated from the 2002-01-26 version of the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji

Nishimura. Original source copyright:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
All rights reserved

The complete text of licenses and disclaimers associated with the the original sources enumerated above at the time of code translation are in the LICENSE.txt file.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own

copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names

of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law

(such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE COMMONS MATH DERIVATIVE WORKS:

The Apache commons-math library includes a number of subcomponents whose implementation is derived from original sources written in C or Fortran. License terms of the original sources are reproduced below.

=====
For the lmdcr, lmpar and qrsolv Fortran routine from minpack and translated in the LevenbergMarquardtOptimizer

class in package
org.apache.commons.math.optimization.general
Original source copyright and license statement:

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF

ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES.

=====

Copyright and license statement for the odex Fortran routine developed by E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff:

Copyright (c) 2004, Ernst Hairer

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright and license statement for the original lapack fortran routines translated in EigenDecompositionImpl class in package org.apache.commons.math.linear:

\$COPYRIGHT\$

Additional copyrights may follow

\$HEADERS\$

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright and license statement for the original Mersenne twister C routines translated in MersenneTwister class in package org.apache.commons.math.random:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.4 commons-collections 3.2.2

1.4.1 Available under license :

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.5 jeromq 0.3.5

1.5.1 Available under license :

GNU General Lesser Public License (LGPL) version 3.0
<http://www.gnu.org/licenses/lgpl-3.0.html>

1.6 annotations 13.0

1.6.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2006 Sascha Weinreuter
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
jar/org/intellij/lang/annotations/Identifier.java
* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
jar/org/intellij/lang/annotations/Pattern.java
* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
jar/org/intellij/lang/annotations/Language.java
*
/opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
jar/org/intellij/lang/annotations/PrintFormat.java
* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
jar/org/intellij/lang/annotations/RegExp.java
* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-
```


jar/org/intellij/lang/annotations/Subst.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2000-2013 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-jar/org/jetbrains/annotations/Contract.java

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-jar/org/intellij/lang/annotations/Flow.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2000-2009 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-jar/org/jetbrains/annotations/Nls.java

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-jar/org/jetbrains/annotations/Nullable.java

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/jetbrains/annotations/NonNls.java

*

/opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/jetbrains/annotations/PropertyKey.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2000-2012 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/jetbrains/annotations/TestOnly.java

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/intellij/lang/annotations/JdkConstants.java

* /opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/jetbrains/annotations/NotNull.java

*

/opt/cola/permits/173667507_1695324641.7896898/0/annotations-13-0-sources-

jar/org/intellij/lang/annotations/MagicConstant.java

1.7 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava

1.8 failureaccess 1.0.1

1.9 j2objc-annotations 1.3

1.9.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

1.10 commons-digester 2.1

1.10.1 Available under license :

Apache Commons Digester

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

1.11 Icms 2.9

1.11.1 Available under license :

Little CMS

Copyright (c) 1998-2011 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

THIS LICENSE APPLIES ONLY TO iccjpeg.c file

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2013, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

The IJG distribution formerly included code to read and write GIF files.

To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

1.12 bean-validation-api 2.0.1.Final

1.12.1 Available under license :

Bean Validation API

License: Apache License, Version 2.0

See the license.txt file in the root directory or <<http://www.apache.org/licenses/LICENSE-2.0>>.

1.13 giflib 5.2.1

1.13.1 Available under license :

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.14 event-stream 1.0.1

1.14.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1015861006_1591858433.32/0/event-stream-1-0-1-sources-jar/META-INF/maven/com.github.malkomich/event-stream/pom.xml

1.15 scala 2.12.10

1.15.1 Available under license :

Scala includes the JLine library:

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO

EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the ASM library.

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Scala

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

Scala includes software developed at LAMP/EPFL (<https://lamp.epfl.ch/>) and Lightbend, Inc. (<https://www.lightbend.com/>).

Licensed under the Apache License, Version 2.0 (the "License"). Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This software includes projects with other licenses -- see `doc/LICENSE.md`.
not-a-legal-formal-parameter-tuple.scala:2: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.

```
    Either create a single parameter accepting the Tuple2,  
    or consider a pattern matching anonymous function: `{ case (a, b) => ... }`  
val x: ((Int, Int) => Int) = (((a, b)) => a)  
      ^
```

not-a-legal-formal-parameter-tuple.scala:3: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.

```
    Either create a single parameter accepting the Tuple2,  
    or consider a pattern matching anonymous function: `{ case (param1, param2) => ... }`  
val y: ((Int, Int, Int) => Int) = (((a, !)) => a)  
      ^
```

not-a-legal-formal-parameter-tuple.scala:4: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.

```
    Either create a single parameter accepting the  
    Tuple3,  
    or consider a pattern matching anonymous function: `{ case (param1, ..., param3) => ... }`  
val z: ((Int, Int, Int) => Int) = (((a, NotAPatternVariableName, c)) => a)  
      ^
```

three errors found

```
{% if site.thisScalaVersion != site.latestScalaVersion %}  
<div class="version-notice">This is the specification of a previous version of Scala. See the <a href="{ {  
site.baseurl} }"/>{ { site.latestScalaVersion } }"/>Scala {{ site.latestScalaVersion }} spec</a>.</div>  
{% endif %}
```

(The MIT License)

Copyright (c) 2013 Greg Allen

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Scala is licensed under the [Apache License Version 2.0](<https://www.apache.org/licenses/LICENSE-2.0>).

Scala License

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Other Licenses

This software includes projects with the following licenses, which are also included in the `licenses/`` directory:

[Apache License](<http://www.apache.org/licenses/LICENSE-2.0.html>)

This license is used by the following third-party libraries:

* jansi

[BSD License](http://www.opensource.org/licenses/bsd-license.php)

This license is used by the following third-party libraries:

- * jline

[BSD 3-Clause License](http://opensource.org/licenses/BSD-3-Clause)

This license is used by the following third-party libraries:

- * asm

[MIT License](http://www.opensource.org/licenses/MIT)

This license is used by the following third-party libraries:

- * jquery

- * tools tooltip

Copyright (c) 2006, Ivan Sagalaev

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the Tools Tooltip library:

Copyright (c) 2009 Tero Piirainen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(c) 2012-2014 GitHub

When using the GitHub logos, be sure to follow the GitHub logo guidelines (<https://github.com/logos>)

Font License: SIL OFL 1.1 (<http://scripts.sil.org/OFL>)

Applies to all font files

Code License: MIT (<http://choosealicense.com/licenses/mit/>)

Applies to all other files

Scala includes the jQuery library:

Copyright (c) 2010 John Resig

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Scala includes the JLine library, which includes the Jansi library.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%)
or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is
based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of

Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.16 error_prone_annotations 2.3.4

1.16.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/Immutable.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/ForOverride.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/Var.java
*
/opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/IncompatibleModifiers.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-
jar/com/google/errorprone/annotations/RequiredModifiers.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2017 The Error Prone Authors.
 *
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
*

/opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/CheckReturnValue.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/NoAllocation.java
*

/opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2016 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/FormatString.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/DoNotMock.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/CompatibleWith.java
*
/opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/RestrictedApi.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/MustBeClosed.java
* /opt/ws_local/PERMITS_SQL/1059106420_1592475238.99/0/error-prone-annotations-2-3-4-sources-1-jar/com/google/errorprone/annotations/FormatMethod.java
```

1.17 jctools-core 3.1.0

1.17.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <artifactId>jctools-core</artifactId>
  <groupId>org.jctools</groupId>
  <version>3.1.0</version>
```

```

<name>Java Concurrency Tools Core Library</name>
<description>Java Concurrency Tools Core Library</description>
<packaging>bundle</packaging>

<dependencies>
<dependency>
<groupId>org.hamcrest</groupId>
<artifactId>hamcrest-all</artifactId>
<version>${hamcrest.version}</version>
<scope>test</scope>
</dependency>

<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>

<dependency>
<groupId>com.google.guava</groupId>
<artifactId>guava-testlib</artifactId>
<version>${guava-testlib.version}</version>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.0.0-M3</version>
<configuration>
<includes>
<include>*</include>
</includes>
</configuration>
</plugin>
<plugin>
<groupId>org.apache.felix</groupId>
<artifactId>maven-bundle-plugin</artifactId>
<version>4.2.1</version>
<extensions>>true</extensions>
<configuration>
<instructions>
<Import-Package>sun.misc;resolution:=optional</Import-Package>
</instructions>
</configuration>
</plugin>

```



```

</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>3.2.0</version>
  <executions>
    <execution>
      <id>attach-sources</id>
      <phase>verify</phase>
      <goals>
        <goal>jar-no-fork</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>3.1.1</version>
  <configuration>
    <additionalOptions>
      <additionalOption>-Xdoclint:none</additionalOption>
    </additionalOptions>
    <source>8</source>
  </configuration>
  <executions>
    <execution>
      <id>attach-javadocs</id>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
</plugins>
</build>

<distributionManagement>
  <repository>
    <id>bintray-jctools-jctools</id>
    <name>jctools-jctools</name>
    <url>https://api.bintray.com/maven/jctools/jctools/jctools-core/?publish=1</url>
  </repository>
</distributionManagement>

<url>https://github.com/JCTools</url>
<inceptionYear>2013</inceptionYear>

```

```
<licenses>
  <license>
    <name>Apache
License, Version 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
    <distribution>repo</distribution>
  </license>
</licenses>

<scm>
  <url>https://github.com/JCTools/JCTools</url>
  <connection>scm:git:https://github.com/JCTools/JCTools</connection>
  <tag>HEAD</tag>
</scm>

<developers>
  <developer>
    <url>https://github.com/nitsanw</url>
  </developer>
  <developer>
    <url>https://github.com/mjpt777</url>
  </developer>
  <developer>
    <url>https://github.com/RichardWarburton</url>
  </developer>
  <developer>
    <url>https://github.com/kay</url>
  </developer>
  <developer>
    <url>https://github.com/franz1981</url>
  </developer>
</developers>

<prerequisites>
  <maven>3.5.0</maven>
</prerequisites>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <java.version>1.6</java.version>
  <java.test.version>1.8</java.test.version>

  <maven.compiler.source>${java.version}</maven.compiler.source>
  <maven.compiler.target>${java.version}</maven.compiler.target>
  <maven.compiler.testSource>${java.test.version}</maven.compiler.testSource>
  <maven.compiler.testTarget>${java.test.version}</maven.compiler.testTarget>

  <hamcrest.version>1.3</hamcrest.version>
```

```
<junit.version>4.12</junit.version>
<guava-testlib.version>21.0</guava-testlib.version>
</properties>
</project>
```

Found

in path(s):

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/META-INF/maven/org.jctools/jctools-core/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/spec/Ordering.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/atomic/SpSCUnboundedAtomicArrayQueue.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/atomic/SpmcAtomicArrayQueue.java

*

/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/MessagePassingQueueUtil.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/atomic/MpmcAtomicArrayQueue.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/util/UnsafeJvmInfo.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/atomic/LinkedQueueAtomicNode.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/SpSCLinkedQueue.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/MpscUnboundedXaddArrayQueue.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org.jctools/queues/spec/Preference.java

* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-

jar/org/jctools/queues/atomic/MpscAtomicArrayQueue.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MessagePassingQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/SupportsIterator.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpmcUnboundedXaddArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpscGrowableArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/BaseSpSCLinkedAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/AtomicQueueFactory.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpmcArrayQueue.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/SpSCAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/SpSCLinkedAtomicQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/SpMCArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/maps/AbstractEntry.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpmcUnboundedXaddChunk.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpscChunkedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/maps/NonBlockingHashMapLong.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/MpscChunkedAtomicArrayQueue.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/ConcurrentSequencedCircularArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/util/UnsafeLongArrayAccess.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpscBlockingConsumerArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/BaseMpscLinkedAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/maps/NonBlockingHashMap.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/BaseMpscLinkedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/MpscGrowableAtomicArrayQueue.java

*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/maps/NonBlockingIdentityHashMap.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/BaseLinkedListQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/util/InternalAPI.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/maps/NonBlockingHashSet.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/util/Pow2.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/spec/ConcurrentQueueSpec.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/atomic/SpSCChunkedAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/maps/ConcurrentAutoTable.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/MpscUnboundedXaddChunk.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/atomic/MpscUnboundedAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/LinkedListNode.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/SpSCUnboundedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/util/PortableJvmInfo.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/ConcurrentCircularArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/maps/NonBlockingSetInt.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/atomic/BaseLinkedListAtomicQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/atomic/SequencedAtomicReferenceArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/MpscLinkedListQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/util/UnsafeRefArrayAccess.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/MpscCompoundQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/SpSCChunkedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-jar/org/jctools/queues/SpSCArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-

```

jar/org/jctools/queues/atomic/AtomicReferenceArrayQueue.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/QueueFactory.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpscArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/util/UnsafeAccess.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/MpscUnboundedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/SpSCGrowAtomicArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/IndexedQueueSizeUtil.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/atomic/MpscLinkedAtomicQueue.java
*
/opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/SpSCGrowArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/util/RangeUtil.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815924_1601014602.33/0/jctools-core-3-1-0-sources-
jar/org/jctools/queues/BaseSpSCLinkedArrayQueue.java

```

1.18 okio 2.8.0

1.18.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2018 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

Found in path(s):

- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/ByteString.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/-Utf8.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Util.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Platform.kt
- *
- /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-Platform.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSink.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/SegmentedByteString.kt
- * /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSource.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Okio.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSource.kt
*
/opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Source.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Sink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Buffer.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/Buffer.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Timeout.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/PeekSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedOkio.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUpgrade.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Throttler.kt
*
/opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUtf8.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSource.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/InflaterSource.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/AsyncTimeout.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSink.kt

*

/opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Source.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentPool.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentPool.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSource.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Segment.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSink.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Sink.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/JvmOkio.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Buffer.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Timeout.kt

*

/opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSource.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSource.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/DeflaterSink.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSink.kt

* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSink.kt

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright (C) 2016 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Options.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Pipe.kt
```

1.19 xerces-j 2.12.2

1.19.1 Available under license :

Apache XML Commons Resolver
Copyright 2006 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation <http://www.apache.org/>

Portions of this code are derived from classes placed in the
public domain by Arbortext on 10 Apr 2000. See:
http://www.arbortext.com/customer_support/updates_and_technical_notes/catalogs/docs/README.htm

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own

copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law

(such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
=====  
== NOTICE file corresponding to section 4(d) of the Apache License, ==  
== Version 2.0, in this case for the Apache Xerces Java distribution. ==  
=====
```

Apache Xerces Java
Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:
- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.

- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xalan Java distribution. ==
=====

Apache Xalan (Xalan serializer)
Copyright 1999-2012 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation.,
<http://www.lotus.com>.
- software copyright (c) 2001-2002, Sun Microsystems.,
<http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation.,
<http://www.ibm.com>.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed

under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.20 httpcomponents-client 4.5.13

1.20.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

=====

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <<http://mozilla.org/MPL/2.0/>>

Mozilla Public License Version 2.0

=====

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code

Form to which the initial Contributor has attached

the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible,

whether at the time of the initial grant or subsequently,
any and
all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- (a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- (b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those

licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free,

non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)

Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise

exploit its

Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer

for sale, have made, import, and otherwise transfer either its

Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License.

Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software;

or

(b) for infringements caused by: (i) Your and any other third party's modifications

of Covered Software, or (ii) the combination of its

Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if

permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- (a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- (b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a

recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```

*
*
* 6. Disclaimer of Warranty
* -----
*
* Covered Software is provided under this License on an "as is"
* basis, without warranty of any kind, either expressed, implied, or
* statutory, including, without limitation, warranties that the
* Covered Software is
* free of defects, merchantable, fit for a
* particular purpose or non-infringing. The entire risk as to the
* quality and performance of the Covered Software is with You.
* Should any Covered Software prove defective in any respect, You
* (not any Contributor) assume the cost of any necessary servicing,
* repair, or correction. This disclaimer of warranty constitutes an

```

* essential part of this License. No use of any Covered Software is *
* authorized under this License except under this disclaimer. *
* *

* *
* 7. Limitation of Liability *
* ----- *

* *
* Under no circumstances and under no legal theory, whether tort *
* (including negligence), contract, or otherwise, shall any *
* Contributor, or anyone who distributes Covered Software as *
* permitted above, be liable to You for any direct, indirect, *
* special, incidental, or consequential damages of any character *
* including, without limitation, damages for lost profits, loss of *
* goodwill, work stoppage, computer failure or malfunction, or any *
* and all other commercial damages or losses, even if such party *
* shall have been informed of the possibility of such damages. This *
* limitation of liability shall not apply to liability for death or *
* personal injury resulting from such party's negligence to the *
* extent applicable law prohibits such limitation. Some *
* jurisdictions do not allow the exclusion or limitation
of *
* incidental or consequential damages, so this exclusion and *
* limitation may not apply to You. *
* *

8. Litigation -----

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous -----

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation

which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Apache HttpComponents Client
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

1.21 guava 30.1-jre

1.21.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/MoreObjects.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/Quantiles.java
```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManagerBridge.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Java8Usage.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2006 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/AppendableWriter.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```
jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/FluentFuture.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/CharEscaperBuilder.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/annotations/VisibleForTesting.java
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2013 The Guava Authors
```

*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashingInputStream.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Verify.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharSequenceReader.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableBiMap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/IndexedImmutableSet.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

```

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This following method is a modified version of one found in
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
* which contained the following notice:
*
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
*
the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
*
* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
*/

```

Found in path(s):

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/MoreExecutors.java

```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeySetMultimap.java

```

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeyListMultimap.java

```

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

jar/com/google/common/collect/AllEqualOrdering.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompactHashMap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingDeque.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RangeMap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeRangeMap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedSet.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultimap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableAsList.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EvictingQueue.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableSet.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractNavigableMap.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2013 The Guava Authors

*

```
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/MoreFiles.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
*
```

```
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Serialization.java
```

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableBiMap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Table.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/EmptyImmutableListMultimap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CollectPreconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/StandardRowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Range.java

```

*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapKeySet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Shorts}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* <http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp>, which contained the

```
* following header:
*
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby disclaims
copyright to this source code.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CharSink.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/Stats.java
```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/html/package-info.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/PairedStatsAccumulator.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/Element.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/io/CharSource.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/package-info.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/StatsAccumulator.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/escape/package-info.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/MutableTypeToInstanceMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/SipHashFunction.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/TypeCapture.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/LongAddable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/io/ByteSource.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/ChecksumHashFunction.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/Invokable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/util/concurrent/RateLimiter.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/base/StandardSystemProperty.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/AbstractByteHasher.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/AbstractInvocationHandler.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/io/BaseEncoding.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/reflect/TypeToInstanceMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/PairedStats.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CartesianList.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/FileWriteMode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/xml/package-
info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManager.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Traverser.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ClosingFuture.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby
disclaims copyright to this source code.
 */
```

Found in path(s):

```
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Hashing.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
 * {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
 * under GWT but with a slightly different signature.
 */
```


*
* <p>We can't use { @code PartiallyGwtIncompatible } because then the GWT compiler
wouldn't recognize
* it as a { @code GwtIncompatible } annotation. And for { @code Futures.catching }, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/SuccessorsFunction.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/InsecureRecursiveDeleteException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MutableGraph.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Network.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2005 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.

/*

* Written by Doug Lea with assistance from members of JCP JSR-166

* Expert Group and released to the public domain, as explained at

* <http://creativecommons.org/publicdomain/zero/1.0/>

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAdder.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/LongAdder.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/Striped64.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Striped64.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Doubles.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableFutureTask.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Bytes.java
```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/internal/Finalizer.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/TrieParser.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MinMaxPriorityQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSequentialIterator.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingListMultimap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/RowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedMapDifference.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DenseImmutableTable.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFuture.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedBytes.java
- *
- /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HostSpecifier.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/SignedBytes.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/GwtIncompatible.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SparseImmutableTable.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/UrlEscapers.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ReferenceEntry.java
- *
- /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MapMakerInternalMap.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeResolver.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Platform.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MapMaker.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheBuilder.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Cut.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataInput.java
- *
- /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SettableFuture.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Callables.java
- * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```
jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/html/HtmlEscapers.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/ArrayBasedCharEscaper.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/Escapers.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
```

```

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
 *   ImmutableSortedMap#of(Comparable, Object)}.</b>
 */

```

Found in path(s):

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```

```

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
 * version.
 */

```

Found in path(s):

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```



```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableClassToInstanceMap.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ArrayTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TableCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableSortedSet.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
```

jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ComparisonChain.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/EmptyImmutableSetMultimap.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ContiguousSet.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Monitor.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/package-
info.java
*

*

/* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedLists.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Ascii.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Strings.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/annotations/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/annotations/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Equivalence.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/Beta.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningExecutorService.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/GwtTransient.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/PatternCompiler.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CommonMatcher.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CommonPattern.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/JdkPattern.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableValueGraph.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractGraph.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MoreCollectors.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableValueGraph.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ElementOrder.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ValueGraphBuilder.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/GraphConstants.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractValueGraph.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/UndirectedGraphConnections.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RangeGwtSerializationDependencies.java

```

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMapMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/DirectedGraphConnections.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MapIteratorCache.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableGraph.java
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2020 The Guava Authors
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
```

```
* in compliance with the License. You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software distributed under the License
```

```
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express
```

```
* or implied. See the License for the specific language governing permissions and limitations under
```

```
* the License.
```

```
*/
```

```
/**
```

```
* Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular
```

* version.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/DoublesMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/IntsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either

* express or implied. See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedAsList.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CollectSpliterators.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Resources.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-


```

jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Function.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Files.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/package-info.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Objects.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

```

jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Closeables.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/package-
info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/AllowConcurrentEvents.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Functions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LittleEndianDataOutputStream.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ByteStreams.java

```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/FarmHashFingerprint64.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AggregateFutureState.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 */
```

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactHashing.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multimaps.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/BiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/Multisets.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ForwardingQueue.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ForwardingConcurrentMap.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ReverseNaturalOrdering.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/SortedSetMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/NullsFirstOrdering.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/SingletonImmutableSet.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/EnumHashBiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/Multiset.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ForwardingListIterator.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/Ordering.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/TreeMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/LinkedListMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/AbstractSortedSetMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ConcurrentHashMultiset.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/HashMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/AbstractSetMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/ImmutableList.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/NullsLastOrdering.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/AbstractBiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/AbstractListMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompoundOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSet.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractMapBasedMultiset.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ArrayListMultimap.java
*

```

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LexicographicalOrdering.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MapDifference.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EnumBiMap.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingLock.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractHashFunction.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableLongArray.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingCondition.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Platform.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Internal.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you

* may not use this file except in compliance with the License. You may

* obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Streams.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Java8Compatibility.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/ToDoubleRounder.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/BigDecimalMath.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/IgnoreJRERRequirement.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/ImmutableSupplier.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionSequencer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

```
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*
* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
* java.util.HashMap
class.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Striped.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/GeneralRange.java
*
```

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Count.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMultiset.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInts.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/WrappingExecutorService.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractScheduledService.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedExecutionException.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/Weigher.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheStats.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractCompositeHashFunction.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/ForwardingLoadingCache.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Optional.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Present.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/PrimitiveSink.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingExecutorService.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TreeRangeSet.java

```

```

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Absent.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Hasher.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Murmur3_128HashFunction.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Enums.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/RemovalCause.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/package-
info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

```

jar/com/google/common/base/FunctionalEquivalence.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/util/concurrent/AsyncFunction.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/Cache.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/BloomFilterStrategies.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/BigIntegerMath.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/AbstractCache.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/util/concurrent/ExecutionError.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/IntMath.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/HashingOutputStream.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/EmptyContiguousSet.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/DoubleMath.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/Funnels.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/CacheLoader.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/hash/Funnel.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/ForwardingCache.java
 *
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/Queues.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/LongMath.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/math/DoubleUtils.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/collect/AbstractSortedMultiset.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/cache/LoadingCache.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/primitives/ParseRequest.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
 jar/com/google/common/util/concurrent/FutureCallback.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/hash/BloomFilter.java

*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/hash/HashFunction.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

/**

* Returns an array containing all of the elements in the specified collection. This method

* returns the elements in the order they are returned by the collection's iterator. The returned

* array is "safe" in that no references to it are maintained by the collection. The caller is

* thus free to modify the returned

array.

*

* <p>This method assumes that the collection size doesn't change while the method is running.

*

* <p>TODO(kevinb): support concurrently modified collections?

*

* @param c the collection for which to return an array of elements

*/

Found in path(s):

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/ObjectArrays.java

1.22 httpcomponents-core 4.4.13

1.22.1 Available under license :

Apache HttpCore

Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.23 httpcore-nio 4.4.13

1.23.1 Available under license :

Apache HttpCore NIO
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.24 httpclient-cache 4.5.13

1.24.1 Available under license :

Apache HttpClient Cache
Copyright 2010-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.25 commons-logging 1.2

1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2001-2006 The Apache Software Foundation.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/LogFactory.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2001-2004,2006 The Apache Software Foundation.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/AvalonLogger.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2004 The Apache Software Foundation.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/impl/WeakHashtable.java

No license file was found, but licenses were detected in source scan.

2004 The Apache Software Foundation.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Found in path(s):

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/package.html

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/impl/package.html

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2001-2004 The Apache Software Foundation.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/LogConfigurationException.java

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/impl/Log4JLogger.java

* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-

jar/org/apache/commons/logging/impl/LogFactoryImpl.java


```
*
/opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/Jdk13LumberjackLogger.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/SimpleLog.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/Jdk14Logger.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/LogSource.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/NoOpLog.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/LogKitLogger.java
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/Log.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2005 The Apache Software Foundation.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
jar/org/apache/commons/logging/impl/ServletContextCleaner.java
```

1.26 commons-validator 1.7

1.26.1 Available under license :

Apache Commons Validator
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.27 commons-io 2.8.0

1.27.1 Available under license :

Apache Commons IO
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.28 commons-codec 1.14

1.28.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)

the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication

on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names

of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law

(such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your

sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Codec

Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from <http://aspell.net/test/orig/batch0.tab>.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

=====

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at <http://stevemorse.org/phoneticinfo.htm>
with permission from the original authors.

Original source copyright:

Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.29 javassist 3.27.0-GA

1.29.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
 *
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/Proxy.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Visitor.java
*
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Member.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/reflect/CannotInvokeException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/CtNewWrappedConstructor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/LoaderClassPath.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/rmi/AppletServer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtConstructor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/rmi/ObjectNotFoundException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/ArrayInit.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/Mnemonic.java
*
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtNewClass.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/SignatureAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtArray.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformNewClass.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/Parser.java
```


* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/DefineClassHelper.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/SubroutineScanner.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtMember.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ExceptionTable.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/ConstructorCall.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/StubGenerator.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/FieldInfo.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/CodeGen.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/CodeAnalyzer.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/IntQueue.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/runtime/DotClass.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/URLClassPath.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/ProxyObjectInputStream.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/InstructionPrinter.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/NestMembersAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/CannotReflectException.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/scopedpool/ScopedClassPoolFactoryImpl.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/LocalVariableAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/CallExpr.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/AttributeInfo.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/KeywordTable.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/CodeIterator.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/SymbolTable.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/Subroutine.java

```

* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/runtime/Inner.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/DefinePackageHelper.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/LineNumberAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/scopedpool/ScopedClassPoolFactory.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/Modifier.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/AnnotationsWriter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/CannotCompileException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/Expr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformFieldAccess.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtPrimitiveType.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/stackmap/BasicBlock.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/Callback.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/MethodInfo.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/JvstCodeGen.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/NewExpr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/stackmap/TypeData.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/ClassFile.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtMethod.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/CondExpr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/scopedpool/ScopedClassPool.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/NewExpr.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/reflect/CannotCreateException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/EnclosingMethodAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/BinExpr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/SourceFileAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/SecurityActions.java

```

* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassMap.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/TypeChecker.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/FieldDecl.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CodeConverter.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/ClassMetaobject.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/NestHostAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/AccessFlag.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/ASTree.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/StackMap.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/NotFoundException.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/Metalevel.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Variable.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/Sample.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/Instanceof.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ClassFilePrinter.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Stmnt.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/framedump.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/annotation/AnnotationImpl.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/AssignExpr.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/Bytecode.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ConstantAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/runtime/Desc.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/ProxyFactory.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/FactoryHelper.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtNewMethod.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-

jar/javassist/tools/rmi/RemoteException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/InstanceOfExpr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Declarator.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtClassType.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/NoSuchClassError.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtBehavior.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Expr.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/ByteArray.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/rmi/RemoteRef.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/stackmap/TypedBlock.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Keyword.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/MemberCodeGen.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/Descriptor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/reflect/Reflection.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/ClassFileWriter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/Frame.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/reflect/Compiler.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/FramePrinter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/DoubleConst.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/scopedpool/ScopedClassPoolRepositoryImpl.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformAfter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/Analyzer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/SerializedProxy.java

* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/MemberResolver.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/StringL.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/Util.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ParameterAnnotationsAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/web/BadHttpRequest.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/BadBytecode.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassPool.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/Proxy.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtNewWrappedMethod.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/Type.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/SyntheticAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/ObjectImporter.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/ExprEditor.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ProceedHandler.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ExceptionsAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/convert/TransformAccessArrayField.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/AnnotationDefaultAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/scopedpool/SoftValueHashMap.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/Loader.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtField.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/Dump.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/MethodDecl.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/CompileError.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/Handler.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ByteStream.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-

```

jar/javassist/bytecode/stackmap/Tracer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/HotSwapAgent.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/FieldAccess.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/InnerClassesAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/web/Viewer.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/ProxyObject.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformCall.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/Symbol.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/ByteArrayClassPath.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/AccessorMaker.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/Lex.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/ASTList.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/MethodCall.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/SerialVersionUID.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformBefore.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/runtime/Cflow.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/Transformer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/AnnotationsAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/LocalVariableTypeAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/DuplicateMemberException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/MethodFilter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/MultiArrayType.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/tools/reflect/Metaobject.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/stackmap/MapMaker.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/JvstTypeChecker.java

```

* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/convert/TransformReadField.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/Loader.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/StackMapTable.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassPoolTail.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassPath.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/HotSwapper.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/Javac.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/ControlFlow.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/Opcode.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/Executor.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/web/Webserver.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/LongVector.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/ProxyObjectOutputStream.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/CastExpr.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/DeprecatedAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtClass.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/Translator.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/Sample.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/CodeAttribute.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/NewArray.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassClassPath.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/IntConst.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/MethodHandler.java
 *
 /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/convert/TransformNew.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/scopedpool/ScopedClassPoolRepository.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/Cast.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/NoFieldException.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/SyntaxError.java
 * /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-

```
jar/javassist/compiler/TokenId.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/RuntimeSupport.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Pair.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformWriteField.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/CtNewConstructor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/ConstPool.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/MultiType.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Javassist, a Java-bytecode translator toolkit.
* Copyright (C) 2004 Bill Burke. All Rights Reserved.
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License. Alternatively, the contents of this file may be used under
* the terms of the GNU Lesser General Public License Version 2.1 or later,
* or the Apache License Version 2.0.
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/MemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/EnumMemberValue.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/MemberValueVisitor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/FloatMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/IntegerMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
```



```

jar/javassist/bytecode/annotation/StringMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/ArrayMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/BooleanMemberValue.java
*
/opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/LongMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/Annotation.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/ShortMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/annotation/DoubleMemberValue.java

```

1.30 zt 1.14

1.30.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-
jar/org/zeroturnaround/zip/extra/AsiExtraField.java
*

```

/opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/ZipLong.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/ZipExtraField.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/ZipShort.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/ExtraFieldUtils.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/ZipConstants.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/extra/UnrecognizedExtraField.java
No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/commons/FileExistsException.java
*
/opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/commons/IOUtils.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/commons/FileUtilsV2_2.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/commons/StringBuilderWriter.java
* /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/commons/FilenameUtils.java
No license file was found, but licenses were detected in source scan.

/**

* Copyright (C) 2012 ZeroTurnaround LLC <support@zeroturnaround.com>
*

- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZTFileUtil.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZipInfoCallback.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/NameMapper.java
- *
- /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/FileSource.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZipEntryUtil.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/timestamps/PreJava8TimestampStrategy.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/Zips.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/timestamps/TimestampStrategy.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ByteSource.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/timestamps/Java8TimestampStrategy.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZipEntrySource.java
- *
- /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/IdentityNameMapper.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZipUtil.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/timestamps/TimestampStrategyFactory.java
- * /opt/cola/permits/1135879776_1613624027.25/0/zt-zip-1-14-sources-1-jar/org/zeroturnaround/zip/ZipEntryCallback.java

1.31 httpcomponents-mime 4.5.12

1.31.1 Available under license :

Apache HttpClient Mime
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.32 commons-codec 1.15

1.32.1 Available under license :

Apache Commons Codec
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from <http://aspell.net/test/orig/batch0.tab>.
Copyright (C) 2002 Kevin Atkinson (kevin@gnu.org)

=====

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at <http://stevemorse.org/phoneticinfo.htm> with permission from the original authors.

Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.33 joda-time 2.10.2

1.33.1 Available under license :

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by
Joda.org (<https://www.joda.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.34 snappy-java 1.1.8.4

1.34.1 Available under license :

This product includes software developed by Google
Snappy: <http://code.google.com/p/snappy/> (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common <http://hadoop.apache.org/>
(Apache 2.0 license)

This library contained statically linked libstdc++. This inclusion is allowed by
"GCC Runtime Library Exception"
<http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html>

== Contributors ==

- * Tatu Saloranta
 - * Providing benchmark suite
- * Alec Wysoker
 - * Performance and memory usage improvement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)

the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication

on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law

(such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.35 Iz4 1.9.3

1.35.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in

certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General
Public License instead of this License.

Copyright (c) 2014, Ipsantil
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This repository uses 2 different licenses :

- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPLv2 license, unless explicitly stated otherwise

Relevant license is reminded at the top of each source file,
and with presence of COPYING or LICENSE file in associated directories.

This model is selected to emphasize that
files in the `lib` directory are designed to be included into 3rd party applications,
while all other files, in `programs`, `tests` or `examples`,
receive more limited attention and support for such scenario.

LZ4 Library

Copyright (c) 2011-2016, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: liblz4

Upstream-Contact: Yann Collet <Cyan4973@github.com>

Source: <https://github.com/lz4/lz4>

Files: *

Copyright: (C) 2011-2020 Yann Collet

License: GPL-2+

The full text of license: <https://github.com/lz4/lz4/blob/dev/lib/LICENSE>

1.36 asm 9.1

1.36.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-
jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-
jar/org/objectweb/asm/RecordComponentVisitor.java
*
/opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-
jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Context.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/TypeReference.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-
jar/org/objectweb/asm/MethodTooLargeException.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ModuleVisitor.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-
jar/org/objectweb/asm/ClassTooLargeException.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ClassWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Attribute.java
*
/opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ClassVisitor.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/AnnotationWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Constants.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ConstantDynamic.java
*
/opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Handle.java
```


* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Opcodes.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Frame.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/SymbolTable.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/ModuleWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/MethodVisitor.java
*
/opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/RecordComponentWriter.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/Symbol.java
* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/CurrentFrame.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF

* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):

* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/package.html

* /opt/cola/permits/1146079232_1616457494.45/0/asm-9-1-sources-jar/org/objectweb/asm/signature/package.html

1.37 Lombok 1.18.20

1.37.1 Available under license :

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
Licenses for included components:

org.ow2.asm:asm
org.ow2.asm:asm-analysis
org.ow2.asm:asm-commons
org.ow2.asm:asm-tree
org.ow2.asm:asm-util
ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors

may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

rzwitserloot/com.zwitserloot.cmdreader

Copyright 2010 Reinier Zwitserloot.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

rzwitserloot/lombok.patcher

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.38 lombok-utils 1.18.12

1.38.1 Available under license :

Copyright (C) 2009-2015 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.39 commons-lang3 3.12.0

1.39.1 Available under license :

Apache Commons Lang
Copyright 2001-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.40 commons-compress 1.21

1.40.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Compress

Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

The files in the package `org.apache.commons.compress.archivers.sevenz`
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (<http://www.7-zip.org/sdk.html>)

The test file `lbzip2_32767.bz2` has been copied from `libbzip2`'s source
repository:

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2019 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented;
you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must
not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

1.41 free-type 2.10.4+dfsg-1

1.41.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: FreeType

Upstream-Contact: freetype-devel@nongnu.org

Source: <https://www.freetype.org/download.html>

Files-Excluded: docs/reference/assets/javascripts

Comment: The files in this path have been removed because they are minified
JavaScript files and are incompatible with the DFSG.

Files: *

Copyright: 1996-2020 David Turner, Robert Wilhelm and Werner Lemberg

1996-2020 Just van Rossum

2002-2020 Roberto Alameda

2003 Huw D M Davies for Codeweavers

2003-2020 Masatake Yamato, Redhat K.K.

2004-2020 Albert Chin-A-Young

2004-2020 Suzuki Toshiya

2007 Dmitry Timoshkov for Codeweavers

2007-2020 Rahul Bhalerao <rahul.bhalerao@redhat.com>

2007-2020 Derek Clegg, Michael Toftdal

2009-2020 Oran Agra, Mickey Gabel

2010-2020 Joel Klinghed

License: GPL-2+ or FTL

Files: vms_make.com

Copyright:

1996-2020 David Turner, Robert Wilhelm and Werner Lemberg

2001, 2002 Francesco Zappa Nardelli

2010-2020 Joel Klinghed

License: FTL and MIT

Files: builds/amiga/*

Copyright: 2005-2020 Werner Lemberg and Detlef Wrkner

License: FTL

Files: builds/amiga/src/base/*

Copyright: 1996-2020 David Turner, Robert Wilhelm, Werner Lemberg and Detlef Wrkner

License: FTL

Files: builds/cmake/FindHarfBuzz.cmake

Copyright: 2012 Intel Corporation

License: BSD-3-Clause

Files: builds/mac/ftmac.c

include/freetype/ftmac.h

src/base/ftmac.c

Copyright: 1996-2020 Just van Rossum, David Turner, Robert Wilhelm and Werner Lemberg

License: FTL

Files: builds/mac/README

Copyright: 2013 Suzuki Toshiya, Leonard Rosenthol, Just van Rossum

License: FTL

Files: builds/unix/aclocal.m4

Copyright: 1992-2018 Free Software Foundation, Inc.

License: FSFULLR and GPL-2+

Files: builds/unix/ax_compare_version.m4

Copyright: 2008 Tim Toolan

License: FSFAP

Files: builds/unix/ax_prog_python_version.m4

Copyright:

2009 Francesco Salvestrini

License: FSFAP

Files: builds/unix/config.guess builds/unix/config.sub

Copyright: 1992-2020 Free Software Foundation, Inc.

License: Permissive and GPL-3+

Files: builds/unix/configure

Copyright: 1992-2018 Free Software Foundation, Inc.

License: FSFUL and GPL-2+

Files: builds/unix/install-sh

Copyright: 1994 X Consortium

License: MIT

Files: builds/unix/ltmain.sh

Copyright: 1996-2015 Free Software Foundation, Inc.

License: GPL-2+ and GPL-3+

Files: builds/unix/pkg.m4

Copyright: 2004 Scott James Remnant <scott@netsplit.com>

License: GPL-2+

Files: debian/*

Copyright: 1996-2019 Christoph Lameter <clameter@waterf.org>, Anthony Fok <foka@debian.org>, Steve Langasek <vorlon@debian.org>, et al.
2018-2020 Hugh McMaster <hugh.mcmaster@outlook.com>

License: GPL-2+

Files: docs/DOCGUIDE

src/sfnt/sfwoff2.c
src/sfnt/woff2tags.c src/sfnt/woff2tags.h

Copyright: 1996-2020 Nikhil Ramakrishnan, David Turner, Robert Wilhelm and Werner Lemberg

License: FTL

Files: docs/INSTALL.CROSS

Copyright: 2006-2020 Suzuki Toshiya, David Turner, Robert Wilhelm and Werner Lemberg

License: FTL

Files: docs/reference/*

Copyright: The FreeType Project

License: FTL

Files: docs/reference/assets/fonts/font-awesome.css

Copyright: 2016 Dave Gandy

License: MIT

Files: docs/reference/assets/fonts/material-icons.css

docs/reference/assets/fonts/specimen/MaterialIcons-Regular.ttf
docs/reference/assets/fonts/specimen/MaterialIcons-Regular.woff
docs/reference/assets/fonts/specimen/MaterialIcons-Regular.woff2

Copyright: 2015 Google, Inc.

License: Apache-2.0

Files: docs/reference/assets/fonts/specimen/FontAwesome.ttf

docs/reference/assets/fonts/specimen/FontAwesome.woff
docs/reference/assets/fonts/specimen/FontAwesome.woff2

Copyright: Dave Gandy

License: OFL-1.1

Files: ft2demos/graph/gblender.h

Copyright: 2001-2020 David Turner

License: FTL

Files: ft2demos/graph/beos/grbeos.cpp

Copyright:

2001-2020 Michael Pfeiffer

License: FTL

Files: ft2demos/graph/mac/grmac.c
Copyright: 1999-2020 Just van Rossum, Antoine Leca,
David Turner, Robert Wilhelm and Werner Lemberg
License: FTL

Files: ft2demos/graph/win32/grwin32.c ft2demos/graph/win32/grwin32.h
ft2demos/graph/x11/grx11.c ft2demos/graph/x11/grx11.h
Copyright: 1999-2020 Antoine Leca, David Turner, Robert Wilhelm and Werner Lemberg
License: FTL

Files: ft2demos/graph/graph.h
ft2demos/graph/grdevice.h
ft2demos/graph/grobjs.h
ft2demos/graph/grtypes.h
Copyright: 1999-2020 The FreeType Development Team
License: FTL

Files: ft2demos/graph/grswizzle.c
Copyright: None
License: Public-Domain
"this filtering code is explicitly placed in the public domain !!"

Files: ft2demos/mac/ftoldmac.c
Copyright: 1996-2020 Suzuki Toshiya, David Turner, Robert Wilhelm and Werner Lemberg
License: FTL

Files: ft2demos/mac/getargv.c
Copyright: 1991-2020 Stichting Mathematisch
Centrum
License: MIT

Files: ft2demos/src/mlgetopt.c ft2demos/src/mlgetopt.h
Copyright: None
License: Public-Domain
"This code is hereby expressly placed in the public domain."

Files: ft2demos/src/ftinspect/*
Copyright: 2016-2020 Werner Lemberg
License: FTL

Files: ft2docs/docs/js/jquery.ba-resize.min.js
Copyright: 2010 Ben Alman
License: MIT or GPL-2+

Files: ft2docs/docs/js/jquery-1.11.0.min.js
Copyright: 2005, 2014 jQuery Foundation, Inc.
License: MIT

Files: ft2docs/docs/tutorial/example2.cpp
ft2docs/docs/tutorial/example3.cpp
ft2docs/docs/tutorial/example4.cpp

Copyright: None

License: Public-Domain

"Public domain."

Files: ft2docs/docs/tutorial/example5.cpp

Copyright: 2016-2018 Static Jobs LLC

License: MIT

Files: include/freetype/ftbzip2.h

src/bzip2/*

Copyright: 2010-2020 Joel Klinghed

License: FTL

Files: include/freetype/ftcid.h

include/freetype/internal/services/svcid.h

src/base/ftcid.c

Copyright: 2007-2020 Derek Clegg

and Michael Toftdal

License: FTL

Files: include/freetype/ftgxval.h

include/freetype/internal/services/svgxval.h

include/freetype/internal/services/svttcmap.h

src/base/ftgxval.c

Copyright: 2003-2020 Masatake Yamato, Red Hat K.K.,

David Turner, Robert Wilhelm and Werner Lemberg

License: FTL

Files: include/freetype/internal/fthash.h

src/base/fthash.c

Copyright: 2000 Computing Research Labs, New Mexico State University

2001-2015 Francesco Zappa Nardelli

License: MIT

Files: include/freetype/internal/ftfork.h

src/base/ftfork.c

Copyright: 2004-2020 Masatake Yamato, Redhat K.K.

License: FTL

Files: src/sfnt/ttcmapc.h

Copyright: 2009-2020 Oran Agra and Mickey Gabel

License: FTL

Files: include/freetype/internal/services/svttglyf.h

src/base/ftpatent.c
src/lzw/ftzopen.c src/lzw/ftzopen.h
Copyright: 2002-2020 David Turner
License: FTL

Files: src/autofit/afindic.c src/autofit/afindic.h
Copyright: 2007-2020 Rahul
Bhalerao
License: FTL

Files: src/base/ftbase.h
Copyright: 2008-2020 David Turner, Robert Wilhelm, Werner Lemberg and Suzuki Toshiya
License: FTL

Files: src/base/fterrors.c
Copyright: 2018-2020 Armin Hasitzka, David Turner, Robert Wilhelm and Werner Lemberg
License: FTL

Files: src/base/md5.c src/base/md5.h
Copyright: None
License: Public-Domain

"No copyright is claimed, and the software is hereby placed in the public domain."

Files: src/bdf/bdf.c
src/bdf/bdfdrivr.c src/bdf/bdfdrivr.h
src/bdf/bdferror.h
src/bdf/module.mk
src/bdf/README
src/bdf/rules.mk
src/pcf/module.mk
src/pcf/pcf.c src/pcf/pcf.h
src/pcf/pcfdrivr.c src/pcf/pcfdrivr.h
src/pcf/pcfread.c src/pcf/pcfread.h
src/pcf/pcfutil.h
src/pcf/README
src/pcf/rules.mk
Copyright: 2000-2014 Francesco Zappa Nardelli
License: MIT

Files: src/bdf/bdf.h src/bdf/bdflib.c
Copyright: 2000 Computing Research Labs, New Mexico State University

2001-2014 Francesco Zappa Nardelli
License: MIT

Files: src/gxvalid/*
Copyright: 2004-2020 Suzuki Toshiya, Masatake Yamato, Red Hat K.K.,
David Turner, Robert Wilhelm and Werner Lemberg

License: FTL

Files: src/gxvalid/gxvfgenc.c

Copyright: 2004-2020 Masatake Yamato, Redhat K.K.

License: FTL

Files: src/gzip/adler32.c

src/gzip/infblock.c src/gzip/infblock.h

src/gzip/infcodes.c src/gzip/infcodes.h

src/gzip/inflate.c

src/gzip/inftrees.c src/gzip/inftrees.h

src/gzip/infutil.c src/gzip/infutil.h

Copyright: 1995-2002 Mark Adler

License: Zlib

Files: src/gzip/ftzconf.h

src/gzip/zutil.c src/gzip/zutil.h

Copyright: 1995-2002 Jean-loup Gailly

License: Zlib

Files: src/gzip/zlib.h

Copyright: 1995-2002 Jean-loup Gailly and Mark Adler

License: Zlib

Files: src/lzw/ftlzw.c src/lzw/rules.mk

Copyright: 2004-2020 Albert Chin-A-Young

License: FTL

Files: src/pcf/pcfutil.c

Copyright: 1990, 1994, 1998 The Open Group

License: OpenGroup-BSD-like

Files:

src/psaux/psarrst.c src/psaux/psarrst.h

src/psaux/psblues.c src/psaux/psblues.h

src/psaux/pserror.c src/psaux/pserror.h

src/psaux/psfixed.h

src/psaux/psfont.c src/psaux/psfont.h

src/psaux/psft.c src/psaux/psft.h

src/psaux/psglue.h

src/psaux/pshints.c src/psaux/pshints.h

src/psaux/psintrap.c src/psaux/psintrap.h

src/psaux/psread.c src/psaux/psread.h

src/psaux/psstack.c src/psaux/psstack.h

src/psaux/pstypes.h

Copyright: 2006-2014 Adobe Systems Incorporated

License: FTL

Files: src/sfnt/pngshim.c src/sfnt/pngshim.h

Copyright: 2013-2020 Google, Inc.

License: FTL

Files: src/sfnt/ttsbit.c

Copyright: 2005-2020 David Turner, Robert Wilhelm and Werner Lemberg

2013 Google, Inc.

License: FTL

Files: src/tools/apinames.c src/tools/chktrcmp.py

Copyright: None

License: Public-Domain

"This code is explicitly placed into the public domain."

Files: src/tools/update-copyright-year

Copyright: 2015-2020

Werner Lemberg

License: FTL

Files: src/tools/ftrandom/ftrandom.c

Copyright: 2005, 2007, 2008, 2013 George Williams

License: BSD-3-Clause

Files: src/truetype/ttgxvar.c src/truetype/ttgxvar.h

Copyright: 2004-2020 David Turner, Robert Wilhelm, Werner Lemberg and George Williams

License: FTL

Files: src/type42/t42drivr.c src/type42/t42drivr.h

src/type42/t42objs.c src/type42/t42objs.h

src/type42/t42parse.c src/type42/t42parse.h

src/type42/t42types.h

Copyright: 2002-2020 Roberto Alameda

License: FTL

Files: src/winfonts/winfnt.c

Copyright: 1996-2020 David Turner, Robert Wilhelm and Werner Lemberg

2003 Huw D M Davies for Codeweavers

2007 Dmitry Timoshkov for Codeweavers

License: FTL

Files: src/winfonts/winfnt.h

Copyright: 1996-2020 David Turner, Robert Wilhelm and Werner Lemberg

2007 Dmitry Timoshkov for Codeweavers

License: FTL

License: Apache-2.0

Licensed under the Apache License, Version 2.0 (the "License");

you may not

use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

On Debian GNU/Linux systems, the complete text of the Apache License Version 2.0 can be found in ``usr/share/common-licenses/Apache-2.0'`.

License: BSD-3-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: FSFAP

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved. This file is offered as-is, without any warranty.

License: FSFUL

This file is free software; the Free Software Foundation gives unlimited permission to copy, distribute and modify it.

License: FSFULLR

This file is free software; the Free Software Foundation gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

License: FTL

The FreeType Project LICENSE

.

2006-Jan-27

.

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

.

.

Introduction

=====

.

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

.

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. (as is' distribution)

.

- o You can use this software for whatever you want, in parts or full form, without having to pay us. (royalty-free' usage)

.

- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge

somewhere in your documentation that you have used the FreeType code. (credits)

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
""""
Portions of this software are copyright <year> The FreeType
Project (www.freetype.org). All rights reserved.
""""
```

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms

=====

0. Definitions

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project`, `FreeType Engine`, `FreeType library`, or `FreeType Distribution`.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

There are two mailing lists related to FreeType:

- o freetype@nongnu.org

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- o freetype-devel@nongnu.org

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

- o <https://www.freetype.org>

License: GPL-2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

On Debian GNU/Linux systems, the complete text of the GNU General Public License version 2 can be found in `~/usr/share/common-licenses/GPL-2'`.

License: GPL-3+

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

On Debian GNU/Linux systems, the complete text of the GNU General Public License version 3 can be found in `~/usr/share/common-licenses/GPL-3'`.

License: MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License: OFL-1.1

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

.
PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

.
The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

.
DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

.
"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

.
"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

.
"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

.
"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.
.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
- 4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
- 5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

License: OpenGroup-BSD-like

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice
and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

License: Permissive

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

License: Zlib

This software
is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

1.42 namespace 1.4.01

1.42.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)

the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication

on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names

of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law

(such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your

sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
=====  
== NOTICE file corresponding to section 4(d) of the Apache License, ==  
== Version 2.0, in this case for the Apache xml-commons xml-apis ==  
== distribution. ==  
=====
```

Apache XML Commons XML APIs
Copyright 1999-2009 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2000 World Wide Web Consortium, <http://www.w3.org>

xml-commons/java/external/LICENSE.dom-documentation.txt \$Id: LICENSE.dom-documentation.txt 226215
2005-06-03 22:49:13Z mrglavas \$

This license came from: <http://www.w3.org/Consortium/Legal/copyright-documents-20021231>

W3C DOCUMENT LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [\$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.
<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle \$Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005)

\$

xml-commons/java/external/LICENSE.dom-software.txt \$Id: LICENSE.dom-software.txt 734314 2009-01-14 03:33:27Z mrglavas \$

This license came from: <http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.zip> (COPYRIGHT.html)

W3C SOFTWARE NOTICE AND LICENSE

Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

Note: The original

version of the W3C Software Copyright Notice and License could be found at <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following

license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms

and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.

Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

xml-commons/java/external/LICENSE.sax.txt \$Id: LICENSE.sax.txt 225954 2002-01-31 23:26:48Z curcuru \$

This license came from: <http://www.megginson.com/SAX/copying.html>

However please note future versions of SAX may be covered under <http://saxproject.org/?selected=pd>

This page is now out of date -- see the new SAX site at <http://www.saxproject.org/> for more up-to-date releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

1.43 alpine-keys 2.4-r1

1.43.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.44 jackson 2.15.0

1.44.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers.

Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.45 log4j-api 2.17.1

1.45.1 Available under license :

Apache Log4j 1.x Compatibility API
Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.46 apache-log4j 2.17.1

1.46.1 Available under license :

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java

Copyright 2005-2006 Tim Fennell

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache license, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the license for the specific language governing permissions and
* limitations under the license.
*/

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.47 log4j-jcl 2.17.1

1.47.1 Available under license :

Apache Log4j Commons Logging Bridge
Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.48 dom 1.0

1.48.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (c) 2000 World Wide Web Consortium,  
* (Massachusetts Institute of Technology, Institut National de  
* Recherche en Informatique et en Automatique, Keio University). All  
* Rights Reserved. This program is distributed under the W3C's Software  
* Intellectual Property License. This program is distributed in the  
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even  
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
* PURPOSE. See W3C License http://www.w3.org/Consortium/Legal/ for more  
* details.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/w3c/dom/html/HTMLDOMImplementation.java  
No license file was found, but licenses were detected in source scan.
```

```
/*  
* Copyright (c) 2001 World Wide Web Consortium,  
* (Massachusetts Institute of Technology, Institut National de  
* Recherche en Informatique et en Automatique, Keio University). All  
* Rights Reserved. This program is distributed under the W3C's Software  
* Intellectual Property License. This program is distributed in the  
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even  
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
* PURPOSE.  
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/apache/xerces/dom3/as/ElementEditAS.java  
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/apache/xerces/dom3/as/ASNotationDeclaration.java  
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/apache/xerces/dom3/as/ASObject.java  
*  
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/apache/xerces/dom3/as/CharacterDataEditAS.java  
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-  
jar/org/apache/xerces/dom3/as/DOMImplementationAS.java  
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
```

```
jar/org/apache/xerces/dom3/as/ASModel.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/DocumentAS.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASAttributeDeclaration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASElementDeclaration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASEntityDeclaration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASNamedObjectMap.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASDataType.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASContentModel.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/DOMASWriter.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/DocumentEditAS.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASObjectList.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/DOMASBuilder.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/DOMASException.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/NodeEditAS.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
* Checks if this content model has had its min/maxOccurs values reduced for
```

```
* purposes of speeding up UPA. If
so, this content model should not be used
* for any purpose other than checking unique particle attribution
*
* @return a boolean that says whether this content has been compacted for UPA
*/
```

Found in path(s):

```
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/models/XSCMValidator.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

```
*/
```

```
/**
```

```
* DOM Level 3 LS CR - Experimental.
* Create a new LSParser. The newly constructed parser may
*
then be configured by means of its DOMConfiguration
* object, and used to parse documents by means of its parse
* method.
* @param mode The mode argument is either
* MODE_SYNCHRONOUS or MODE_ASYNCHRONOUS, if
* mode is MODE_SYNCHRONOUS then the
* LSParser that is created will operate in synchronous
* mode, if it's MODE_ASYNCHRONOUS then the
* LSParser that is created will operate in asynchronous
* mode.
* @param schemaType An absolute URI representing the type of the schema
* language used during the load of a Document using the
* newly created LSParser. Note that no lexical checking
* is done on the absolute URI. In order to create a
* LSParser for any kind of schema types (i.e. the
* LSParser will be free to use any schema
```

found), use the value

- * `<code>null</code>`.
- * `<p >Note:` For W3C XML Schema [`XML Schema Part 1`]
- * , applications must use the value
- * `<code>"http://www.w3.org/2001/XMLSchema"</code>`. For XML DTD [`XML 1.0`],
- * applications must use the value
- * `<code>"http://www.w3.org/TR/REC-xml"</code>`. Other Schema languages
- * are outside the scope of the W3C and therefore should recommend an
- * absolute URI in order to use this method.
- * `@return` The newly created `<code>LSParser</code>` object. This
- * `<code>LSParser</code>` is either synchronous or asynchronous
- * depending on the value of the `<code>mode</code>` argument.
- * `<p >Note:` By default, the newly created `<code>LSParser</code>`
- * does not contain a `<code>DOMErrorHandler</code>`, i.e. the value of
- * the "``
- * `error-handler`" configuration parameter is `<code>null</code>`. However, implementations
- * may provide a default error handler at creation time. In that case,
- * the initial value of the `<code>"error-handler"</code>` configuration
- * parameter on the new created `<code>LSParser</code>` contains a
- * reference to the default error handler.
- * `@exception` `DOMException`
- * `NOT_SUPPORTED_ERR`: Raised if the requested mode or schema type is
- * not supported.
- */

Found in path(s):

- * `/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/CoreDOMImplementationImpl.java`
- No license file was found, but licenses were detected in source scan.

/*

- * Licensed to the Apache Software Foundation (ASF) under one or more
- * contributor license agreements. See the NOTICE file distributed with
- * this work for additional information regarding copyright ownership.
- * The ASF licenses this file to You under the Apache License, Version 2.0
- * (the "License"); you may not use this file except in compliance with
- * the License. You may obtain a copy of the License at
- *
- * `http://www.apache.org/licenses/LICENSE-2.0`
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDocumentInfo.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDElementTraverser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xni/parser/XMLErrorHandler.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLFrameElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/XMLEntityDescriptionImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/ShadowedSymbolTable.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/util/ObjectListImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLImgElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLOptgroupElement.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xni/parser/XMLInputSource.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLImgElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredCDATASectionImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDAbstractIDConstraintTraverser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLTimerElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSParticle.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLWmlElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/AnySimpleDV.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/StringList.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLTimerElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xpath/regex/CaseInsensitiveMap.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/opti/SchemaDOMImplementation.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/opti/DefaultXMLDocumentHandler.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLAnchorElement.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/HTMLSerializer.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLHtmlElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/stax/DefaultNamespaceContext.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xinclude/MultipleScopeNamespaceSupport.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/opti/SchemaParsingConfig.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/XML11NSDTDValidator.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xpointer/XPointerMessageFormatter.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSException.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLMetaElementImpl.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLQuoteElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/EncodingMap.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xinclude/XIncludeTextReader.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMInputImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/ObjectFactory.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xpointer/XPointerHandler.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/XMLCatalogResolver.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/TeeXMLDocumentFilterImpl.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLTableSectionElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/models/CMStateSet.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/impl/dv/xs/Base64BinaryDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/DraconianErrorHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/JAXPValidatorComponent.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTitleElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/XML11NonValidatingConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/ElementDefinitionImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLElementDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSAttributeGroupDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/AbstractDOMParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLSelectElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLLabelElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLFieldSetElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLCardElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/ShortListImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/util/Base64.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/CDATASectionImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/XSGrammarPoolContainer.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/CMBuilder.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLMetaElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/XSSimpleType.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/XSSimpleTypeDelegate.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/SchemaDOMParser.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/RegularExpression.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/ElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/events/UIEventImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/Method.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/EntityReferenceImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/datatypes/ByteList.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLBodyElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLNSDTDValidator.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/XSCMBinOp.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredAttrImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/DOMUtil.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLIElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/SAXParserFactoryImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xpointer/ElementSchemePointer.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/msg/XMLMessageFormatter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/DefaultElement.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/SoftReferenceGrammarPool.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/AbstractXMLSchema.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/AttrNSImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/datatype/DurationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLTdElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLEntityDescription.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLDTDScanner.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/SecurityManager.java

*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSModel.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/XMLErrorCode.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/validation/DOMResultAugmentor.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/identity/FieldActivator.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/ParentNode.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/util/XIntPool.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLTableCellElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/SchemaSymbols.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/ObjectFactory.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/io/UTF16Reader.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/TextImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMStringListImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/parsers/NonValidatingConfiguration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLImageElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DecimalDV.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSModelImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/ShortList.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/dtd/ListDatatypeValidator.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/validation/StAXEventResultBuilder.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredAttrNSImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/DTDGrammar.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredEntityReferenceImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/datatypes/ObjectList.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/identity/XPathMatcher.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLHeadElement.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/PSVLErrorList.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/dtd/XML11NMTOKENDatatypeValidator.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/stax/events/StartDocumentImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/XMLSerializer.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/identity/KeyRef.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSDDescription.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xni/XMLAttributes.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLHeadElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredEntityImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/validation/StAXDocumentHandler.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/io/Latin1Reader.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/LCount.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/XMLDTDProcessor.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/BooleanDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xni/parser/XMLEntityResolver.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMMessageFormatter.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/SchemaGrammar.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xni/grammars/XMLGrammarDescription.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/util/XSInputSource.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/ValidatedInfo.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

```

jar/org/apache/xerces/dom/ChildNode.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/util/XSNamedMapImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/Constants.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/events/MouseEventImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/XMLDTDValidatorFilter.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/XMLDocumentScannerImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/Util.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xinclude/XInclude11TextReader.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/DefaultValidationErrorHandler.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLOptgroupElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/SchemaValidatorConfiguration.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLPostfieldElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/traversers/StAXSchemaParser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DeferredNotationImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/CoreDocumentImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/XMLNSDocumentScannerImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/ReadOnlyGrammarPool.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DOMErrorImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/XMLAttributesImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/traversers/XSDAttributeTraverser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/XMLDocumentParser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/traversers/XSDComplexTypeTraverser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/DatatypeMessageFormatter.java

```

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLIElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLNoopElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/SecuritySupport.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/DatatypeException.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/dtd/ENTITYDatatypeValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLInputElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLTableElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/ElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/SchemaContentHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/SoftReferenceSymbolTableConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/BalancedDTDGrammar.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xpointer/ShortHandPointer.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/BasicParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLPreElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/SecurityConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/AbstractXMLDocumentParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/validation/ValidationState.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/HTMLdtd.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLDOMImplementation.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/AugmentationsImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSImplementation.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

```

jar/org/apache/xerces/xni/parser/XMLDocumentScanner.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/SAXLocatorWrapper.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/SynchronizedSymbolTable.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/XSGrammarBucket.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLNoopElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xpointer/XPointerErrorHandler.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLDTDSources.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/stax/events/ProcessingInstructionImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/NamespacesContext.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/XMLSchemaFactory.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLTdElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLDListElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xs/DayTimeDurationDV.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLUElementImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/util/XSNamedMap4Types.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/datatypes/XSDateTime.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/opti/DefaultText.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/models/CMLLeaf.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/XSTerm.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLComponent.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLSmallElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/DocumentBuilderImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/DatatypeValidator.java

```


* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTextAreaElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/AttributePSVI.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/Token.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredNode.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/XML11Serializer.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDAttributeGroupTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/HexBinaryDV.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLGoElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/SimpleLocator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/MessageFormatter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLErrorReporter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSLoaderImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XML11NamespaceBinder.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/SecuritySupport.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLFormElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/XMLResourceIdentifierImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/XPath.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLMenuElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLCollectionImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLDocumentFragmentHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSComplexTypeDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/SAXInputSource.java

*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/XPointerParserConfiguration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/datatype/SerializedDuration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/io/UTF8Reader.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/InvalidDatatypeValueException.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/DOMInputSource.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/XSNotationDecl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLAreaElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLBigElement.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/XMLChar.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLPostfieldElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DeepNodeListImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/EntityReferenceImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/ProcessingInstructionImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLPrevElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DOMXSImplementationSourceImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/models/MixedContentModel.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/Augmentations.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xs/BaseDVFactory.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/XMLVersionDetector.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/grammars/XMLDTDDescription.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/IDDatatypeValidator.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/StAXValidatorHelper.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/xs/XSComplexTypeDefinition.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/NotationImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLBElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/identity/UniqueOrKey.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/datatype/DatatypeFactoryImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLAnchorElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/DOMSerializerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLIFrameElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/LineSeparator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/XMLLocatorWrapper.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSWildcard.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/StreamValidatorHelper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/StAXLocationWrapper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLDoElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/IntegratedParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLNotationDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/ValidatorHelper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/BaseSchemaDVFactory.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/identity/IdentityConstraint.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/datatypes/XSDouble.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xinclude/SecuritySupport.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/CommentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredProcessingInstructionImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/CommentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLSetvarElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/ASDOMImplementationImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/NotationDeclarationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/SoftReferenceSymbolTable.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/validation/EntityState.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/XHTMLSerializer.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLWmlElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/ASModelImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredDocumentTypeImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/DOMValidatorHelper.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSAnnotationInfo.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XML11DTDSscannerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/NodeImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/XMLGrammarCachingConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLBaseElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xpointer/XPointerPart.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/QName.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLEmElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/grammars/XMLGrammarPool.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLFormControl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredTextImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/impl/dv/xs/FloatDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XML11DocumentScannerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/ErrorHandlerWrapper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSAttributeGroupDefinition.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/models/CMAAny.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/StartElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/DOMSerializer.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSModelGroupDefinition.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/EndElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/QNameDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLBrElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLScriptElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/XMLStringBuffer.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/events/MutationEventImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/ObjectFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDGroupTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/FullDVFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xinclude/ObjectFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/DTDImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/DOMParserImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLTrElement.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/models/CMUniOp.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DocumentFragmentImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLSelectElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDSimpleTypeTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XML11EntityScanner.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/StringDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/IDREFDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLButtonElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/identity/Field.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DocumentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/TextImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLOneventElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/util/ByteListImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/XIncludeParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLStrongElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/CachingParserPool.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/ParserConfigurationSettings.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLDocument.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/EmptyXMLSchema.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSObjectList.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/DefaultNode.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLMapElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XML11NSDocumentScannerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLSmallElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLDTDCContentModelHandler.java

*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLEmElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/PSVDocumentImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/DTDGrammarBucket.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/ElementPSVImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/REUtil.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLParseException.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/Op.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLDTDCContentModelSource.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/XML11DTDCConfiguration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLTemplateElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/SecuritySupport.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLInputElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/NMTOKENDatatypeValidator.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLParagraphElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xml/serialize/OutputFormat.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/NOTATIONDatatypeValidator.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/DTDParse.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/SymbolHash.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xs/ListDV.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLULListElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DOMImplementationSourceImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLBrElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

```

jar/org/apache/xerces/dom/CharacterDataImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/RangeImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xml/serialize/SerializerFactoryImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/XMLSchemaLoader.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/XMLScanner.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/XMLGrammarPoolImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/ElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLStrongElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/XSNotationDeclaration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xml/serialize/EncodingInfo.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLMetaElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/models/CMNodeFactory.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLFontElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLTableElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLAppletElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xs/IDDV.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/SecuritySupport.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/XSObject.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/SchemaDVFactory.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/XMLEntityDecl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/ErrorHandlerProxy.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/DTDDVFactoryImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/models/CMNode.java

```


* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/util/StringListImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLTemplateElement.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/parsers/XML11Configurable.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/InvalidDatatypeFacetException.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDAbstractParticleTraverser.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/validation/StAXStreamResultBuilder.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/datatype/XMLGregorianCalendarImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/util/XInt.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSConstants.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/Serializer.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/SubstitutionGroupHandler.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/TypeValidator.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/SAXParserImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/parsers/XMLParser.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSGroupDecl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DoubleDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/Version.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/IntStack.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/io/ASCIIReader.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSIDCDefinition.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLCardElement.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/WMLFieldsetElement.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/dom/AttrImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLAElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/DefaultDocument.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/FilePathToURI.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/ObjectFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLPrevElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/util/HexBin.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/datatype/SerializedXMLGregorianCalendar.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLDTDContentModelFilter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSModelGroup.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLBaseFontElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLDTDLoader.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredElementDefinitionImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSElementDeclHelper.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLComponentManager.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/PrimeNumberSequenceGenerator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLOneventElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/SchemaDVFactoryImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/ImmutableLocation.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/XMLSchema.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSLoader.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/UnionDV.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XNIException.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLDocumentSource.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/grammars/XSGrammar.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLBElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/SAXParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/datatypes/XSDecimal.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSWildcardDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DOMConfigurationImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTableColElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLTrElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLDOMImplementationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/XSCMUniOp.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/grammars/XMLGrammarLoader.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/validation/ConfigurableValidationState.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/IndentPrinter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/ElementState.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/ObjectFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/PSVIDOMImplementationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/models/DFACContentModel.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/AttributeMap.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/EndDocumentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLPElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/datatypes/XSFloat.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/dtd/StringDatatypeValidator.java

*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/models/XSCMRepeatingLeaf.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/PSVIAttrNSImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/XMLDocumentFragmentScannerImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/ValidationContext.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/grammars/Grammar.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLAccessElementImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/Match.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/XSMessageFormatter.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xml/serialize/TextSerializer.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/stax/events/NamespaceImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/XSAttributeDeclaration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/JAXPConstants.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLAElement.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/EntityResolverWrapper.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DOMOutputImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DOMImplementationImpl.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/SecuritySupport.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/ExternalSubsetResolver.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/NodeListCache.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/PSVIElementNSImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xinclude/XIncludeHandler.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/SAXMessageFormatter.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/dom/DeferredCommentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTableElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLHRElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSNamespaceItemList.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/DOMEntityResolverWrapper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/RegexParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/ElementNSImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/AttributeImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/dtd/IDREFDatatypeValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLUElement.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLHeadElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/XSCMLLeaf.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/LocatorProxy.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/identity/Selector.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/dtd/XML11IDDatatypeValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSElementDeclaration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSAnnotationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/LSInputList.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/XPathException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/EntityDeclarationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/AbstractSAXParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLBigElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLLocator.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSTypeDefinition.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/ItemPSVI.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/URI.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLObjectElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/AnyAtomicDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/NamedNodeMapImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLBuilder.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/WeakReferenceXMLSchema.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/datatypes/XSQName.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/NodeIteratorImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/DefaultErrorHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/BMPattern.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/DTDConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/SimpleXMLSchema.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLLegendElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/YearMonthDurationDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDWildcardTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLHeadingElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/XSFacets.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/ElementPSVI.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLRefreshElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/XSGrammarPool.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLResourceIdentifier.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLDirectoryElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/XML11Configuration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/EntityImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/io/MalformedByteSequenceException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/UnparsedEntityHandler.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLAccessElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/XMLSymbols.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xpointer/XPointerProcessor.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLSetvarElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLDTDHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSMultiValueFacet.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSAnnotation.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLLinkElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/events/EventImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/DTDDVFactory.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/identity/ValueStore.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/ValidatorHandlerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/DOMErrorHandlerWrapper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/stax/events/XMLEventImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/XS10TypeHelper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLDivElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSNamespaceItem.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/util/XML11Char.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLFrameSetElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLNamespaceBinder.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/XMLGrammarPreparser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSImplementationImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLBRElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/RangeExceptionImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/PSVIProvider.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XMLSchemaException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSNamedMap.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLStyleElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSAttributeUseImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/validation/ValidationManager.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/EntityResolver2Wrapper.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSElementDecl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/DOMParser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLAttributeDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLLIElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSParticleDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/models/ContentModelValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/StAXInputSource.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLContentSpec.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DocumentTypeImpl.java
 *

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLOptionElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xinclude/XIncludeMessageFormatter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/AttributePSVImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLSelectElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/XMLString.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/XMLSchemaValidatorComponentManager.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTableRowElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLListElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/SchemaDOM.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLDTDScannerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLInputElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/PrecisionDecimalDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/DOMDocumentHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLPullParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/SecureProcessingConfiguration.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLDocumentFilter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/XSObjectListImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/NamedNodeMapImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xs/XSFacet.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLGoElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDAbstractTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xml/serialize/Printer.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/XML11DTDProcessor.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/XMLDTDDescription.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLRefreshElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xpath/regex/ParserForXMLSchema.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMNormalizer.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/IntegerDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/jaxp/DocumentBuilderFactoryImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/html/dom/HTMLIsIndexElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dtd/models/SimpleContentModel.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredDocumentImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSSimpleTypeDefinition.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/stax/EmptyLocation.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/SerializerFactory.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/HTTPInputSource.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/Encodings.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSAttributeUse.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xml/serialize/BaseMarkupSerializer.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/stax/events/CharactersImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredDOMImplementationImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/stax/XMLEventFactoryImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/JAXPNamespaceContextWrapper.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/util/NamespaceSupport.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

jar/org/apache/xerces/xni/XMLDocumentHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DOMLocatorImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/util/LSInputListImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSDeclarationPool.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/JAXPValidationMessageFormatter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/models/CMBinOp.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/ExtendedSchemaDVFactoryImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XML11DTDValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/EntityDV.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/util/SymbolTable.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/AnyURIDV.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLEntityScanner.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLEntityManager.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/XSAttributeDecl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xinclude/XIncludeNamespaceSupport.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/XSEmptyCM.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLOptGroupElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/DVFactoryException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLDocumentImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/xs/SchemaDateTimeException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLOptionElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLPElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLOptionElementImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/WMLElement.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLParamElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/TreeWalkerImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dv/dtd/XML11DTDDVFactoryImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/ParseException.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/parser/XMLDTDFilter.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/dtd/XMLDTDValidator.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/traversers/XSDNotationTraverser.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/xni/grammars/XMLSchemaDescription.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/jaxp/validation/DOMResultBuilder.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLAnchorElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLModElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLDocumentImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/dom/DeferredElementNSImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/html/dom/HTMLTableCaptionElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/XMLEntityHandler.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/parsers/StandardParserConfiguration.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/models/XSAllCM.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xs/opti/AttrImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLFieldsetElementImpl.java
 *
 /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/wml/dom/WMLDoElementImpl.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
 jar/org/apache/xerces/impl/xpath/regex/RangeToken.java
 * /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-

```

jar/org/apache/xerces/parsers/DOMASBuilderImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/NameNodeListImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLDOMImplementationImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/io/UCSReader.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/XIncludeAwareParserConfiguration.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/RevalidationHandler.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSValue.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/NodeImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/XML11IDREFDatatypeValidator.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLConfigurationException.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/AttributesProxy.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/ValidatorImpl.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/parsers/XMLGrammarParser.java
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/XMLSimpleType.java
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xs/SchemaNamespaceSupport.java
No license file was found, but licenses were detected in source scan.

```

```

/*

```

```

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMImplementationListImpl.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.

* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/**

* used to check the 3 constraints against each complex type

* (should be each model group):

* Unique Particle

Attribution, Particle Derivation (Restriction),

* Element Declarations Consistent.

*/

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSConstraints.java

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/datatypes/package.html

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more

* contributor license agreements. See the NOTICE file distributed with

* this work for additional information regarding copyright ownership.

* The ASF licenses this file to You under the Apache License, Version 2.0

* (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

// Unique Particle Attribution

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/models/XSDFACM.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more

* contributor license agreements. See the NOTICE file distributed with

* this work for additional information regarding copyright ownership.

* The ASF licenses this file to You under the Apache License, Version 2.0

* (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/AbstractDateTimeDV.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/YearDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDKeyrefTraverser.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DateDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DateTimeDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDUniqueOrKeyTraverser.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/MonthDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DurationDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/MonthDayDV.java

*

/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/TimeDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSAttributeChecker.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/YearMonthDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSModelGroupImpl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/XSSimpleTypeDecl.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDHandler.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/DayDV.java

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XMLSchemaValidator.java

No license file was found, but licenses were detected in source scan.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/XMLSchemaMessages.properties


```

* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/message.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/DOMMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/XPointerMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/DatatypeMessages.properties
*
/opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/message_fr.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/SAXMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/XIncludeMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/XMLMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/XMLSerializerMessages.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xpath/regex/message_ja.properties
* /opt/cola/permits/1257218116_1642790294.82/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/msg/JAXPValidationMessages.properties

```

1.49 dagger 2.4

1.49.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2012 Google, Inc.
* Copyright (C) 2012 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Lazy.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Google, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/DelegateFactory.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/MapKey.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/MapFactory.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Component.java

*

/opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/InstanceFactory.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/SingleCheck.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/Factory.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/MapProviderFactory.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/Collections.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/SetFactory.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 Google Inc.

* Copyright (C) 2012 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Provides.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/MembersInjectors.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 Google, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Multibindings.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/package-info.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/IntKey.java

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/StringKey.java

*

/opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/LongKey.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/Beta.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/ClassKey.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Subcomponent.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2016 Google, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Reusable.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/ProviderOfLazy.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Bounds.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/Preconditions.java
*
/opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/internal/DoubleCheck.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-
jar/dagger/multibindings/ElementsIntoSet.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/IntoMap.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/package-info.java
* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/multibindings/IntoSet.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2012 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/Module.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 Square, Inc.

* Copyright (C) 2009 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1257772315_1643010337.24/0/dagger-2-4-sources-jar/dagger/MembersInjector.java

1.50 paranamer 2.5.1

1.50.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

*

* Portions Copyright (c) 2007 Paul Hammant

* Portions copyright (c) 2000-2007 INRIA, France Telecom

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/BytecodeReadingParanamer.java
No license file was found, but licenses were detected in source scan.

/**

*

* Copyright (c) 2009 Paul Hammant

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.
*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/AnnotationParanamer.java

No license file was found, but licenses were detected in source scan.

/**

*

* Copyright (c) 2007 Paul Hammant

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

*

*/

Found in path(s):

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/Paranamer.java

No license file was found, but licenses were detected in source scan.

/**

*

* Copyright (c) 2007 Paul Hammant

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF

* THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/NullParanamer.java

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/AdaptiveParanamer.java

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/ParameterNamesNotFoundException.java

*

/opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/CachingParanamer.java

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/DefaultParanamer.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2007 Paul Hammant

* Copyright 2007 ThinkTank Maths Limited

*
* ThinkTank Maths Limited grants a non-revocable, perpetual licence
* to Paul Hammant for unlimited use, relicensing and redistribution. No
* explicit permission is required from ThinkTank Maths Limited for
* any future decisions made with regard to this file.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1257772269_1643010353.74/0/paranamer-2-5-1-sources-
jar/com/thoughtworks/paranamer/JavadocParanamer.java

1.51 avro 1.11.0

1.51.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Trevni Java Avro
Copyright 2009-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

1.52 annotations 4.1.1.4

1.52.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2012 The Android Open Source Project  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1258811491_1643111603.33/0/annotations-4-1-1-4-sources-  
jar/android/annotation/SuppressLint.java  
* /opt/cola/permits/1258811491_1643111603.33/0/annotations-4-1-1-4-sources-  
jar/android/annotation/TargetApi.java
```

1.53 animal-sniffer-annotation 1.19

1.53.1 Available under license :

No license file was found, but licenses were detected in source scan.

The MIT License

Copyright (c) 2009 codehaus.org.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):

* /opt/cola/permits/1258876677_1643115784.16/0/animal-sniffer-annotations-1-19-sources-jar/META-INF/maven/org.codehaus.mojo/animal-sniffer-annotations/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* The MIT License

*

* Copyright (c) 2008 Kohsuke Kawaguchi and codehaus.org.

*

* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:

*

* The above copyright notice and this permission notice shall be included in

* all copies or substantial portions of the Software.

```
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT
* HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
* THE SOFTWARE.
*
*/
```

Found in path(s):

```
* /opt/cola/permits/1258876677_1643115784.16/0/animal-sniffer-annotations-1-19-sources-
jar/org/codehaus/mojo/animal_sniffer/IgnoreJRERequirement.java
```

1.54 error_prone_annotations 2.3.3

1.54.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-
jar/com/google/errorprone/annotations/CompatibleWith.java
* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-
jar/com/google/errorprone/annotations/FormatMethod.java
* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-
jar/com/google/errorprone/annotations/MustBeClosed.java
*
* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-
jar/com/google/errorprone/annotations/RestrictedApi.java
```


* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/FormatString.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java

* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/DoNotCall.java

* /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java

*

/opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/CheckReturnValue.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/NoAllocation.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2015 The Error Prone Authors.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/CompileTimeConstant.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/Immutable.java
- *
- /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/RequiredModifiers.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/Var.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/ForOverride.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
- * /opt/cola/permits/1264660152_1643871085.34/0/error-prone-annotations-2-3-3-sources-jar/com/google/errorprone/annotations/IncompatibleModifiers.java

1.55 picocontainer 2.15

1.55.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 */
```

Found in path(s):

```
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/ProviderAdapter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/ComponentLifecycle.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/security/CustomPermissionsURLClassLoader.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/ComponentMonitorStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Stored.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/FactoryInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/references/SimpleReference.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Guarded.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/Provider.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/CompositeInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Injector.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/InjectInto.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/references/ThreadLocalMapObjectReference.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/CompositeInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/Injector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
```

jar/org/picocontainer/injectors/MultiInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/ObjectReference.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Decorated.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/Reinjection.java
*

/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/references/ThreadLocalReference.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/Reinjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/MultiInjection.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD style
 * license a copy of which has been included with this distribution in the
 * LICENSE.txt file.
 */
```

Found in path(s):

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/classname/ClassPathElement.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/classname/ClassName.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/classname/ClassLoadingPicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/parameters/DefaultConstructorParameter.java
*

/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/classname/DefaultClassLoadingPicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/parameters/NullParameter.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.      *
 * ----- *
 * The software in this package is published under the terms of the BSD   *
 * style license a copy of which has been included with this distribution in *
 * the LICENSE.txt file. *
 */
```

Found in path(s):

```
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/LifecycleState.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/StartableLifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/visitors/MethodCallingVisitor.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/AbstractMonitoringLifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/visitors/TraversalCheckingVisitor.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/converters/Converter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/ReflectionLifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoVisitor.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/ComponentAdapter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/JavaEE5LifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/Converters.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/LifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/visitors/VerifyingVisitor.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/DefaultLifecycleState.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/CompositeLifecycleStrategy.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/lifecycle/ReflectionLifecycleException.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/visitors/AbstractPicoVisitor.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) PicoContainer Organization. All rights reserved.
* -----
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Original code by
*/
*****/
```

Found in path(s):

- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/AnnotatedFieldInjection.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/NamedFieldInjector.java
- *
- /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/Automated.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/Locked.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/NameBinding.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/AdaptingBehavior.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/PropertyApplicator.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/AnnotatedFieldInjector.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/monitors/LifecycleComponentMonitor.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/PicoContainer.java
- *
- /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/Automating.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/ComponentFactory.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/adapters/AbstractAdapter.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/Injectors.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Characteristics.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/PropertyApplying.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/annotations/Inject.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/containers/TransientPicoContainer.java
- *
- /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/DefaultPicoContainer.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/lifecycle/NullLifecycleStrategy.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/parameters/BasicComponentParameter.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-

jar/org/picocontainer/behaviors/Synchronizing.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/NamedMethodInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoVerificationException.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoException.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/annotations/Cache.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/CommandLinePicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/TypedFieldInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/adapters/InstanceAdapter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/TypedFieldInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/NamedFieldInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/parameters/CollectionComponentParameter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Locking.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/annotations/Nullable.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Behaviors.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/AbstractInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/BehaviorFactory.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoBuilder.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/InjectionFactory.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/AnnotatedMethodInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/ImplementationHiding.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/HiddenImplementation.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/parameters/ComponentParameter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/ImmutablePicoContainer.java

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/Synchronized.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/SetterInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/PicoClassNotFoundException.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Original code by Paul Hammant & Obie Fernandez & Aslak Hellesøy
 */
```

Found in path(s):

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/monitors/NullComponentMonitor.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Original code by Jon Tirsén
 */
```

Found in path(s):

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/AbstractBehavior.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) NanoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Original code by Joerg Schaibe
 */
```


Found in path(s):

- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/Decorating.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/behaviors/FieldDecorating.java

No license file was found, but licenses were detected in source scan.

```

/*****
* Copyright (C) PicoContainer Organization. All rights reserved.      *
* ----- *
* The software in this package is published under the terms of the BSD  *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *
* *
* Original code by Paul Hammaant *
*****/

```

Found in path(s):

- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/monitors/ComponentMonitorHelper.java
- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/monitors/WriterComponentMonitor.java

*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/monitors/ConsoleComponentMonitor.java

No license file was found, but licenses were detected in source scan.

```

/*****
* Copyright (C) PicoContainer Organization. All rights reserved.      *
* ----- *
* The software in this package is published under the terms of the BSD  *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *
* *
* Original code by the committers *
*****/

```

Found in path(s):

- * /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/containers/AbstractDelegatingMutablePicoContainer.java

No license file was found, but licenses were detected in source scan.

```

/*****
* Copyright (C) PicoContainer Organization. All rights reserved.      *
* ----- *
* The software in this package is published under the terms of the BSD  *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *

```

* Original Code By: Centerline Computers, Inc. *

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/injectors/PrimitiveMemberChecker.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) PicoContainer Committers. All rights reserved. *
* ----- *
* The software in this package is published under the terms of the BSD *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *
* *
* Original code by Joerg Schaibe *

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/BindKey.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) PicoContainer Organization. All rights reserved. *
* ----- *
* The software in this package is published under the terms of the BSD *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *
* *
* Original code by Paul Hammant & Obie Fernandez & Aslak *

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/ComponentMonitor.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) PicoContainer Organization. All rights reserved. *
* ----- *
* The software in this package is published under the terms of the BSD *
* style license a copy of which has been included with this distribution in *
* the LICENSE.txt file. *
* *
* Idea by Rachel Davies, Original code by various *

Found in path(s):

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/CompositePicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/MutablePicoContainer.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Original code by Mauro Talevi
 */
```

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/monitors/AbstractComponentMonitor.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 * * Original code by
 */
```

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/AbstractBehaviorFactory.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 */
```

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/PropertiesPicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/MethodInjector.java

```
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Intercepting.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoLifecycleException.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/ForgetfulConstructorInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Intercepted.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/SystemPropertiesPicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/containers/CommandLineArgumentsPicoContainer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/ParameterNameBinding.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/parameters/AbstractParameter.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/SingleMemberInjector.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (c) PicoContainer Organization. All rights reserved.
* -----
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*/
```

```
Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/composers/RegexComposer.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/monitors/ComposingMonitor.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (c) PicoContainer Organization. All rights reserved.
* -----
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Idea by Rachel Davies, Original code by Jon Tirsen
*/
```

Found in path(s):

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/parameters/ConstantParameter.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Parameter.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Original code by Paul Hammant
 */
```

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Behavior.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/containers/EmptyPicoContainer.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the license.html file.
 *
 * Idea by Rachel Davies, Original code by Aslak Hellesoy and Paul Hammant
 */
```

Found in path(s):
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Disposable.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-jar/org/picocontainer/Startable.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) PicoContainer Organization. All rights reserved.
 * -----
 * The software in this package is published under the terms of the BSD
 * style license a copy of which has been included with this distribution in
 * the LICENSE.txt file.
 *
 * Idea by Rachel Davies, Original code by Aslak Hellesoy and Paul Hammant
 */
```

Found in path(s):

```

* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/ThreadCached.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Guarding.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/SetterInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/MethodInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/AnnotatedMethodInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/AdaptingInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/ConstructorInjection.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/ThreadCaching.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/injectors/ConstructorInjector.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Storing.java
*
/opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Caching.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/PicoCompositionException.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/OptInCaching.java
* /opt/cola/permits/1266565840_1644246713.67/0/picocontainer-2-15-sources-
jar/org/picocontainer/behaviors/Cached.java

```

1.56 handy-uri-templates 2.1.8

1.56.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright 2012, Ryan J. McDonough
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

```

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/impl/Operator.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/Expression.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/UriUtil.java
 - *
 - /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/UriTransient.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/impl/Modifier.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/VariableExpansionException.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/VarName.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/VarExploder.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/impl/VarSpec.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/UriTemplate.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/DefaultVarExploder.java
 - *
 - /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/impl/UriTemplateParser.java
 - * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/impl/VarExploderFactory.java
- No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

- * /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/META-INF/maven/com.damnhandy/handy-uri-templates/pom.xml
- No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2013, Ryan J. McDonough
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.

- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

* /opt/cola/permits/1273317603_1645099869.69/0/handy-uri-templates-2-1-8-sources-jar/com/damnhandy/uri/template/MalformedUriTemplateException.java

1.57 javax-annotation-api 1.3.2

1.57.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8.

Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing

Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.

Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under

intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims,

each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee

for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the

License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES

THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS

DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software,

we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any

program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its

scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute

verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code

for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute

the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only

way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These
Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits
your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but

you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

/*

* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*

* Copyright (c) 2005-2018 Oracle and/or its affiliates. All rights reserved.

*

* The contents of this file are subject to the terms of either the GNU

* General Public License Version 2 only ("GPL") or the Common Development

* and Distribution License("CDDL") (collectively, the "License"). You

* may not use this file except in compliance with the License. You can

* obtain a copy of the License at

* <https://oss.oracle.com/licenses/CDDL+GPL-1.1>

* or LICENSE.txt. See the License for the specific

- * language governing permissions and limitations under the License.
- *
- * When distributing the software, include this License Header Notice in each
- * file and include the License file at LICENSE.txt.
- *
- * GPL Classpath Exception:
- * Oracle designates this particular file as subject to the "Classpath"
- * exception as provided by Oracle in the GPL Version 2 section of the License
- * file that accompanied this code.
- *
- *

Modifications:

- * If applicable, add the following below the License Header, with the fields
- * enclosed by brackets [] replaced by your own identifying information:
- * "Portions Copyright [year] [name of copyright owner]"
- *
- * Contributor(s):
- * If you wish your version of this file to be governed by only the CDDL or
- * only the GPL Version 2, indicate your decision by adding "[Contributor]
- * elects to include this software in this distribution under the [CDDL or GPL
- * Version 2] license." If you don't indicate a single choice of license, a
- * recipient has the option to distribute your version of this file under
- * either the CDDL, the GPL Version 2 or to extend the choice of license to
- * its licensees as provided above. However, if you add GPL Version 2 code
- * and therefore, elected the GPL Version 2 license, then the option applies
- * only if the new code is made subject to such option by the copyright
- * holder.
- */

1.58 error_prone_annotations 2.10.0

1.58.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright 2014 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and

```

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/concurrent/LockMethod.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java

*

/opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/NoAllocation.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/RestrictedApi.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/MustBeClosed.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/CompatibleWith.java

*

/opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/DoNotMock.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/FormatMethod.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/FormatString.java

No license file was found, but licenses were detected in source scan.

Copyright 2015 The Error Prone Authors.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/META-INF/maven/com.google.errorprone/error_prone_annotations/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/InlineMeValidationDisabled.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/InlineMe.java

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/Modifier.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/CheckReturnValue.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
*
/opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
*
/opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-jar/com/google/errorprone/annotations/Var.java

```
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-
jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-
jar/com/google/errorprone/annotations/Immutable.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-
jar/com/google/errorprone/annotations/IncompatibleModifiers.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-
jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1287386049_1647249958.14/0/error-prone-annotations-2-10-0-sources-
jar/com/google/errorprone/annotations/ForOverride.java
```

1.59 gson 2.9.0

1.59.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/DateTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/TreeTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/UnsafeAllocator.java
*
/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/sql/SqlDateTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/LazilyParsedNumber.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
```


jar/com/google/gson/internal/sql/SqlTimeTypeAdapter.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonStreamParser.java

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldAttributes.java

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonParser.java

*

/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-

jar/com/google/gson/LongSerializationPolicy.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 The Gson authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-

jar/com/google/gson/internal/JavaVersion.java

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-

jar/com/google/gson/internal/PreJava9DateFormatProvider.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2011 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/JsonTreeReader.java
*
/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/TypeAdapterFactory.java
*
/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

/*
```

* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/LinkedTreeMap.java

No license file was found, but licenses were detected in source scan.

/**

* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/\$Gson\$Types.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonWriter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonToken.java
*
/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonReader.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonScope.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/GsonBuildConfig.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*/

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ToNumberStrategy.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ToNumberPolicy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Streams.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSyntaxException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/annotations/JsonAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-
jar/com/google/gson/internal/bind/NumberTypeAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Until.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonElement.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Excluder.java
- *
- /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ExclusionStrategy.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonParseException.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/\$Gson\$Preconditions.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonIOException.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonObject.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/InstanceCreator.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Primitives.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonNull.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/SerializedName.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonDeserializationContext.java
- *
- /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldNamingPolicy.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonPrimitive.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldNamingStrategy.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/Gson.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSerializer.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSerializationContext.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Since.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonDeserializer.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/reflect/TypeToken.java
- *
- /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/ObjectConstructor.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/DefaultDateTypeAdapter.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Expose.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/GsonBuilder.java
- * /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonArray.java

1.60 guava 31.1-jre

1.60.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Verify.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionContext.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/HashingInputStream.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
No license file was found, but licenses were detected in source scan.
```



```
/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims
 * copyright to this source code.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/SmallCharMatcher.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/MapMaker.java
```

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
```

```

jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/xml/XmlEscapers.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/Escapers.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/SignedBytes.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingFuture.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/TypeResolver.java

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/MapMakerInternalMap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Platform.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/DenseImmutableTable.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/html/HtmlEscapers.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/package-info.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/FinalizablePhantomReference.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/LineBuffer.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Function.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Throwables.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/CountingOutputStream.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/Closeables.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/eventbus/DeadEvent.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Preconditions.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Supplier.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableFuture.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/MultiInputStream.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/Primitives.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/package-info.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Objects.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/LineReader.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Interners.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/eventbus/AsyncEventBus.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/CountingInputStream.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/eventbus/Subscribe.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/package-info.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/FinalizableReference.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/DirectExecutor.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/AbstractIterator.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/ExecutionList.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/eventbus/package-info.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/io/ByteStreams.java
 *
 /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/collect/EnumMultiset.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/FinalizableReferenceQueue.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Functions.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/eventbus/AllowConcurrentEvents.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/collect/HashBiMap.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/io/Resources.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/FinalizableSoftReference.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/io/CharStreams.java
 *
 /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Defaults.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Predicate.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Charsets.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Suppliers.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/FinalizableWeakReference.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/Files.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/AbstractFuture.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/EventBus.java

*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Predicates.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/DescendingMultiset.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredSetMultimap.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredEntryMultimap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableAsList.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CompactLinkedHashSet.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractMultimap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableList.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableMap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TransformedIterator.java

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/DescendingImmutableSortedSet.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractNavigableMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredMultimap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisetBridge.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```

jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TreeRangeMap.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2015 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
* in compliance with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software distributed under the License  
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
* or implied. See the License for the specific language governing permissions and limitations under  
* the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/hash/MacHashFunction.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/hash/LittleEndianByteArray.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/util/concurrent/CombinedFuture.java  
*  
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/io/ReaderInputStream.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/util/concurrent/InterruptibleTask.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/util/concurrent/AggregateFutureState.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/hash/FarmHashFingerprint64.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/util/concurrent/AsyncCallable.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/collect/ConsumingQueueIterator.java  
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-  
jar/com/google/common/util/concurrent/Platform.java  
No license file was found, but licenses were detected in source scan.
```

```
/*  
* Copyright (C) 2013 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");
```


* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/MoreFiles.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntry.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for web specializations of methods of { @code Floats }. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/FloatsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/BaseEncoding.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/Stats.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/StatsAccumulator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/xml/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/SmoothRateLimiter.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/FileWriteMode.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/TypeCapture.java

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/LongAddable.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManager.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/Invokable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/html/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/CharSource.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/ChecksumHashFunction.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/AbstractByteHasher.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/TypeToInstanceMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/CharSink.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/ByteSource.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/reflect/AbstractInvocationHandler.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```

jar/com/google/common/reflect/MutableTypeToInstanceMap.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
 * version.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright (C) 2009 The Guava Authors
```

*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/MutableClassToInstanceMap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Synchronized.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ListMultimap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/EnumBiMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/TreeMultiset.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingQueue.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```

jar/com/google/common/collect/AbstractMapBasedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractSortedSetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Multiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Ordering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractListMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/BiMap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SortedSetMultimap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/MapDifference.java
*

```

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingList.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/EnumHashMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractSetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/CompoundOrdering.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingCollection.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>


```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisets.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned
array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ObjectArrays.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby
disclaims copyright to this source code.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Hashing.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/Traverser.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ClosingFuture.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/AbstractBaseGraph.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/BaseGraph.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you
* may not use this file except in compliance with the License. You may
* obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied. See the License for the specific language governing
* permissions and limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Streams.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/IndexedImmutableSet.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableBiMap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMap.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/BaseImmutableMultimap.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under the License.

*/

/**

* Holder for web specializations of methods of { @code Shorts }. Intended to be empty for regular version.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/ShortsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2005 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/annotations/package-info.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/Atomics.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/package-info.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SortedLists.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/net/package-info.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/annotations/Beta.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ContiguousSet.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Ascii.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Equivalence.java

```
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Strings.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/ImmutableIntArray.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/ImmutableLongArray.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*/
```

```
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Internal.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular
* version.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/DoublesMethodsForWeb.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

```

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/*

* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at
* <http://creativecommons.org/licenses/publicdomain>

*

* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
* `java.util.HashMap`

class.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManagerBridge.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at


```

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
 *   ImmutableSortedMap#of(Comparable, Object)}.</b>
 */

```

Found in path(s):

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```

```

/*
 * Copyright (C) 2011 The Guava Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

```

Found in path(s):

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

```

```

/*
 * Copyright (C) 2009 The Guava Authors
 *

```

```

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
* {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
* under GWT but with a slightly different signature.
*
* <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler
wouldn't recognize
* it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

```

Found in path(s):

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Partially.java

```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/Network.java

```

```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```

```
jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ImmutableGraph.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MutableGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MutableNetwork.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/InsecureRecursiveDeleteException.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionSequencer.java
*
```

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMultiset.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSortedSet.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableAsList.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ComparisonChain.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableClassToInstanceMap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableList.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/TableCollectors.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/DiscreteDomain.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableTable.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableEnumSet.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/EmptyImmutableSetMultimap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ArrayTable.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2008 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableBiMap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapValues.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultimap.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CollectPreconditions.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/StandardRowSortedTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapEntrySet.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableCollection.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Table.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Platform.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Range.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/EmptyImmutableListMultimap.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2021 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/NullnessCasts.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/NullnessCasts.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/NullnessCasts.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedInteger.java
*
```

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/BloomFilter.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Present.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/Crc32HashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/LoadingCache.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/HashCode.java


```

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/Funnel.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionError.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/package-info.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/Hashing.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Optional.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```

jar/com/google/common/base/Enums.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/primitives/UnsignedInts.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/cache/RemovalCause.java
 *
 /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/AbstractCompositeHashFunction.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/primitives/UnsignedLong.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Absent.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/ForwardingExecutorService.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/PrimitiveSink.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/AbstractStreamingHasher.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/cache/ForwardingLoadingCache.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
 *
 /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/Funnels.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/net/MediaType.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/math/MathPreconditions.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/base/Ticker.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/cache/ForwardingCache.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/math/BigIntegerMath.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/Hasher.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/util/concurrent/WrappingExecutorService.java
 *
 /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/net/HostAndPort.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/hash/Murmur3_128HashFunction.java
 * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
 jar/com/google/common/cache/RemovalNotification.java

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Queues.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMapMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ForwardingNetwork.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
```

```

jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/Comparators.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MapRetrievalCache.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/GraphConstants.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/EndpointPair.java

```

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/DirectedNetworkConnections.java
No license file was found, but licenses were detected in source scan.
```

```
// Copyright 2011 Google Inc. All Rights Reserved.
```

```
Found in path(s):
```

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/hash/Fingerprint2011.java
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2019 The Guava Authors
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/CompactHashing.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/graph/IncidentEdgeSet.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2008 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/net/InetAddresses.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/MultiReader.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/collect/FluentIterable.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/net/PercentEscaper.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Doubles.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Chars.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-

```
jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Ints.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/TrieParser.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Stopwatch.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/Converter.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
```

* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular
* version.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/primitives/IntsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/Subscriber.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/MoreObjects.java

*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenerCallQueue.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/math/Quantiles.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/Dispatcher.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberRegistry.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/GeneralRange.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/Count.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/GwtTransient.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/package-info.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/CollectSpliterators.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/BigDecimalMath.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/Java8Compatibility.java
- *

- /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/Java8Compatibility.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/ToDoubleRounder.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (C) 2010 The Guava Authors

*

- * Licensed under the Apache License, Version 2.0 (the "License");

- * you may not use this file except in compliance with the License.

- * You may obtain a copy of the License at

*

- * <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software

- * distributed under the License is distributed on an "AS IS" BASIS,

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

- * See the License for the specific language governing permissions and

- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingListMultimap.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableListIterator.java
- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/RowSortedTable.java
- *

- /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/MinMaxPriorityQueue.java

- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableCollection.java

- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/AbstractSequentialIterator.java

- * /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/SortedMapDifference.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ForwardingSetMultimap.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2021 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/html/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/net/ParametricNullness.java
*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/xml/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/eventbus/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/html/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ElementTypesAreNonnullByDefault.java
*
```

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/cache/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/escape/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/hash/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/math/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/ElementTypesAreNonnullByDefault.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/xml/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/primitives/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/collect/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/eventbus/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/net/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/ParametricNullness.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/escape/ParametricNullness.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/graph/ParametricNullness.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/JdkPattern.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/CommonMatcher.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/base/CommonPattern.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2006 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
*

/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/escape/CharEscaperBuilder.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-
jar/com/google/common/util/concurrent/TimeoutFuture.java

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/annotations/VisibleForTesting.java

*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java

*
/opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/io/PatternFilenameFilter.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

/*

* This following method is a modified version of one found in

* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30>

* which contained the following notice:

*

* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to

*

the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

*

* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.

*/

Found in path(s):

* /opt/cola/permits/1301953781_1649307346.56/0/guava-31-1-jre-sources-1-jar/com/google/common/util/concurrent/MoreExecutors.java

1.61 javapoet 1.13.0

1.61.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-jar/com/squareup/javapoet/LineWrapper.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Google, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/ClassName.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/JavaFile.java

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/AnnotationSpec.java

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/TypeName.java

*
/opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/CodeWriter.java

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/MethodSpec.java

```

* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/TypeVariableName.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/NameAllocator.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/FieldSpec.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/CodeBlock.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-jar/com/squareup/javapoet/Util.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/ParameterizedTypeName.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/TypeSpec.java
*
/opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/ArrayTypeName.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/WildcardTypeName.java
* /opt/cola/permits/1312156089_1650477546.68/0/javapoet-1-13-0-sources-1-
jar/com/squareup/javapoet/ParameterSpec.java

```

1.62 guava 31.0.1-android

1.62.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

```

jar/com/google/common/eventbus/Dispatcher.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/base/MoreObjects.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/ListenerCallQueue.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/eventbus/Subscriber.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2008 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/Tables.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/RegularImmutableMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/ImmutableMapValues.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/StandardRowSortedTable.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/Collections2.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/Platform.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/ImmutableSortedSet.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/Table.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/Range.java

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableCollection.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableBiMap.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/UnmodifiableIterator.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/EmptyImmutableListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/CollectPreconditions.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

```

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ExecutionSequencer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Java8Compatibility.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/ToDoubleRounder.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/BigDecimalMath.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/AtomicLongMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Present.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
*
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/package-info.java
*
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/EmptyContiguousSet.java
```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/Hasher.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/WrappingExecutorService.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/BloomFilter.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/CacheBuilderSpec.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/Murmur3_32HashFunction.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/HashCode.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/ForwardingLoadingCache.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeParameter.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/package-info.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/UnsignedLongs.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/DoubleUtils.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/MediaType.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/RemovalCause.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Optional.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/HashFunction.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/RemovalListener.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/LongMath.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ForwardingExecutorService.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/Murmur3_128HashFunction.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractScheduledService.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractSortedMultiset.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/UnsignedLong.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

```

jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/FunctionalEquivalence.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Enums.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/CacheLoader.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java

```



```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractStreamingHasher.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Absent.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Uninterruptibles.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

```

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/ImmutableIntArray.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingCondition.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for web specializations of methods of { @code Shorts }. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*

```
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/IndexedImmutableSet.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractSequentialIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SortedMapDifference.java
*
```

```
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingSortedSetMultimap.java
```

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RowSortedTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/MinMaxPriorityQueue.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
```

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/Runnables.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/SubscriberExceptionContext.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeVisitor.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/HashingInputStream.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CharSequenceReader.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Utf8.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractTable.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/VerifyException.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/SubscriberExceptionHandler.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Verify.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/FilteredMultimapValues.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

```
/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned
array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ObjectArrays.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/GwtTransient.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
```

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Strings.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Atomic.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/annotations/Beta.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/net/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

```

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Floats }. Intended to be empty for regular
* version.
*/

```

Found in path(s):

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```

```

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/DirectedNetworkConnections.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/UndirectedGraphConnections.java

```


* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Comparators.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/EdgesConnecting.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/NetworkBuilder.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/StandardMutableValueGraph.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/StandardNetwork.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/ForwardingValueGraph.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/ValueGraphBuilder.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/MapIteratorCache.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/StandardMutableGraph.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/NetworkConnections.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/GraphBuilder.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/AbstractDirectedNetworkConnections.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/StandardValueGraph.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/GraphConstants.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/RangeGwtSerializationDependencies.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/EndpointPairIterator.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/ImmutableValueGraph.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/EndpointPair.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/GraphConnections.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

```
jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/UndirectedNetworkConnections.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/ElementOrder.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/AbstractGraph.java
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2016 The Guava Authors
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
```

```
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
```

```
*/
```

```
/**
```

```
* Holder for extra methods of { @code Objects } only in web. Intended to be empty for regular
* version.
```

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/SortedMultisets.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/SortedMultiset.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RangeMap.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FilteredEntryMultimap.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractNavigableMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TreeTraverser.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/CompactHashMap.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/CompactLinkedHashMap.java
*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/TransformedListIterator.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AsyncCallable.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/LittleEndianByteArray.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AggregateFutureState.java

*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/CombinedFuture.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* <http://creativecommons.org/publicdomain/zero/1.0/>
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/Striped64.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/Subscribe.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/MultiInputStream.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Objects.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/FinalizableSoftReference.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/LineReader.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Charsets.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/FinalizableWeakReference.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/LineBuffer.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Predicate.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ExecutionList.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ByteStreams.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Function.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/Files.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/FinalizableReferenceQueue.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/Primitives.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Throwables.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/HashBiMap.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/AllowConcurrentEvents.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/LittleEndianDataInputStream.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Functions.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/Resources.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/FinalizableReference.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/package-info.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/EnumMultiset.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/package-info.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/package-info.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Preconditions.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/DirectExecutor.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/MutableGraph.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/Network.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/ImmutableNetwork.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/MultimapBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/graph/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/*

* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at

* <http://creativecommons.org/licenses/publicdomain>

*

* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's

* `java.util.HashMap`

class.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
 *   ImmutableSortedMap#of(Comparable, Object)}.</b>
 */

```

Found in path(s):

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

```

Found in path(s):

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Platform.java

```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/html/package-info.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/ByteSink.java
*
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ServiceManager.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/AbstractByteHasher.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/Closer.java
*
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/xml/package-info.java
```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/ClassPath.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CharSink.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/package-info.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/ChecksumHashFunction.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeCapture.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/package-info.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/FilteredKeyMultimap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/FileWriteMode.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/Invokable.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableRangeSet.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/LongAddable.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/CharSource.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/MutableTypeToInstanceMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeToInstanceMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/LinearTransformation.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/SmoothRateLimiter.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/RateLimiter.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/LongAddables.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ByteSource.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/StatsAccumulator.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/Stats.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/AbstractInvocationHandler.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ObjectCountLinkedHashMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ClosingFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/AbstractBaseGraph.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ObjectCountHashMap.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
```

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This following method is a modified version of one found in
* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30>
* which contained the following notice:
*
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
*
the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>
*
* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/MoreExecutors.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2008 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/Floats.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/SequentialExecutor.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/Shorts.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/Ints.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Chars.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/thirdparty/publicsuffix/TrieParser.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Converter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Booleans.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/FileBackedOutputStream.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableSortedMap.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ArrayTable.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableSetMultimap.java
 - *
 - /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/RegularImmutableSortedSet.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableTable.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ComputationException.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractIndexedListIterator.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/EmptyImmutableSetMultimap.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableEnumSet.java
 - *
 - /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ComparisonChain.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableAsList.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/SingletonImmutableTable.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/RegularImmutableList.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/DiscreteDomain.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
 - * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingTable.java
- No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
 * {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
 * under GWT but with a slightly different signature.
 *
 * <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler
 * wouldn't recognize
 * it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
 * compiler to autostrip the normal server method in order to expose the special, inherited GWT
 * version.
 */
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Partially.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ExplicitOrdering.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Multiset.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/LexicographicalOrdering.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractIterator.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractSortedSetMultimap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/UsingToStringOrdering.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/RegularImmutableSet.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingObject.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingQueue.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/EnumBiMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/EnumHashBiMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ConcurrentHashMultiset.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/NullsFirstOrdering.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableList.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/HashMultiset.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Multimap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/CompoundOrdering.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractSetMultimap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractMapBasedMultiset.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ByFunctionOrdering.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/SortedSetMultimap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

```

jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingSet.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingMultiset.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractListMultimap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SetMultimap.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/MapDifference.java

```

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/NullsLastOrdering.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Ordering.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ImmutableSet.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/TreeMultiset.java
 *
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Sets.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingCollection.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Synchronized.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ListMultimap.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ReverseOrdering.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/AbstractMapBasedMultimap.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Multimaps.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ArrayListMultimap.java
 *
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingSortedSet.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingConcurrentMap.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/TreeMultimap.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ForwardingIterator.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ClassToInstanceMap.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Iterators.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Maps.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ComparatorOrdering.java
 *
 /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/LinkedHashMultiset.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Interner.java
 * /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Iterables.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2020 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ServiceManagerBridge.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2011 The Guava Authors.  
*  
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
* in compliance with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software distributed under the License  
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
* or implied. See the License for the specific language governing permissions and limitations under  
* the License.  
*/
```

Found in path(s):
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2008 The Guava Authors
```

```
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
* following header:
*
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby
disclaims copyright to this source code.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Hashing.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/PatternCompiler.java
```

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/CommonPattern.java
```


* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/JdkPattern.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/CommonMatcher.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under the License.

*/

/*

* This method was rewritten in Java from an intermediate step of the Murmur hash function in <http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp>, which contained the following header:

*

* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author hereby disclaims

copyright to this source code.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/SmallCharMatcher.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/MapMakerInternalMap.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/HostSpecifier.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeResolver.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/Platform.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/annotations/GwtIncompatible.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/annotations/GwtCompatible.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractIdleService.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/ReferenceEntry.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/UnsignedBytes.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ForwardingFuture.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/ArrayBasedCharEscaper.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ByteProcessor.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/html/HtmlEscapers.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/UrlEscapers.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/Cut.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/Callables.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/InternetDomainName.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ByteArrayDataInput.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/ForwardingListenableFuture.java

*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/escape/Escapers.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/util/concurrent/AbstractService.java

*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/SignedBytes.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2006 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/FluentFuture.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java

*

/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/TimeLimiter.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/FakeTimeLimiter.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/escape/CharEscaperBuilder.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/AggregateFuture.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2021 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/util/concurrent/NullnessCasts.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/base/NullnessCasts.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-

jar/com/google/common/collect/NullnessCasts.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.
*/
/**
* Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular
* version.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/ParametricNullness.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/html/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/ParametricNullness.java

* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/ParametricNullness.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/hash/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/escape/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/xml/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/html/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/net/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/xml/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/math/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ParametricNullness.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/primitives/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/eventbus/ParametricNullness.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/util/concurrent/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/graph/ParametricNullness.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/reflect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/collect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/io/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/base/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-jar/com/google/common/cache/ElementTypesAreNonnullByDefault.java
No license file was found, but licenses were detected in source scan.

/*

```
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular
* version.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/Count.java
*
/opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RangeSet.java
```



```
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1312950193_1650560245.64/0/guava-31-0-1-android-sources-1-
jar/com/google/common/collect/AbstractRangeSet.java
```

1.63 animal-sniffer-annotation 1.21

1.63.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* The MIT License
*
* Copyright (c) 2008 Kohsuke Kawaguchi and codehaus.org.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT
* HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
* THE SOFTWARE.
*/
```

Found in path(s):

```
* /opt/cola/permits/1319211632_1651231549.68/0/animal-sniffer-annotations-1-21-sources-
jar/org/codehaus/mojo/animal_sniffer/IgnoreJRERequirement.java
```

No license file was found, but licenses were detected in source scan.

The MIT License

Copyright (c) 2009 codehaus.org.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):

* /opt/cola/permits/1319211632_1651231549.68/0/animal-sniffer-annotations-1-21-sources-jar/META-INF/maven/org.codehaus.mojo/animal-sniffer-annotations/pom.xml

1.64 ion-java 1.0.2

1.64.1 Available under license :

Amazon Ion Java

Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.65 gson 2.8.9

1.65.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Gson authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* **WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND**, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/GsonBuildConfig.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/TreeTypeAdapter.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/DateTypeAdapter.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/ConstructorConstructor.java

*

*/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/sql/SqlDateTypeAdapter.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/UnsafeAllocator.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/sql/SqlTimeTypeAdapter.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LazilyParsedNumber.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/TypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/TypeAdapter.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/JsonTreeWriter.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonElement.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonObject.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonArray.java
- *
- /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/FieldNamingStrategy.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/SerializedName.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonDeserializationContext.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/Excluder.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/FieldNamingPolicy.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/\$Gson\$Preconditions.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/DefaultDateTypeAdapter.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonSerializationContext.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonParseException.java
- *
- /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/ObjectConstructor.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/Gson.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonIOException.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/reflect/TypeToken.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonDeserializer.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/Expose.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/Primitives.java
- * /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/GsonBuilder.java
- *

```
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/annotations/Until.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/InstanceCreator.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonNull.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/annotations/JsonAdapter.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
*/
```

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LinkedHashMap.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LinkedTreeMap.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 Google Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonReader.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/MalformedJsonException.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonScope.java

*

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonToken.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonWriter.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 The Gson authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/PreJava9ReflectionAccessor.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/ReflectionAccessor.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/UnsafeReflectionAccessor.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/PreJava9DateFormatProvider.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/JavaVersion.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/NumberTypeAdapter.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2021 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/ToNumberPolicy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/ToNumberStrategy.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/internal/Streams.java

No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2008 Google Inc.
*
*/

```
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
*/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/internal/$Gson$Types.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
*/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/JsonStreamParser.java
```

```
*/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/LongSerializationPolicy.java
```

```
*/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-
jar/com/google/gson/FieldAttributes.java
```

```
*/
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonParser.java
```

1.66 perfmark-api 0.25.0

1.66.1 Available under license :

Apache-2.0

1.67 zstd-jni 1.5.2-1

1.67.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.github.luben.zstd_jni

Bnd-LastModified: 1642851452470

Bundle-Description: JNI bindings for Zstd native library that provides fast and high compression lossless algorithm for Java and all JVM languages.

Bundle-License: <https://opensource.org/licenses/BSD-2-Clause>;description=BSD 2-Clause License

Bundle-ManifestVersion: 2

Bundle-Name: zstd-jni

Bundle-NativeCode: darwin/x86_64/libzstd-jni-1.5.2-1.dylib;osname=MacOS;osname=MacOSX;processor=x86_64, darwin/aarch64/libzstd-jni-1.5.2-1.dylib;osname=MacOS;osname=MacOSX;processor=aarch64, freebsd/amd64/libzstd-jni-1.5.2-1.so;osname=FreeBSD;processor=amd64, freebsd/i386/libzstd-jni-1.5.2-1.so;osname=FreeBSD;processor=i386, linux/aarch64/libzstd-jni-1.5.2-1.so;osname=Linux;processor=aarch64, linux/amd64/libzstd-jni-1.5.2-1.so;osname=Linux;processor=amd64, linux/arm/libzstd-jni-1.5.2-1.so;osname=Linux;processor=arm, linux/i386/libzstd-jni-1.5.2-1.so;osname=Linux;processor=i386, linux/mips64/libzstd-jni-1.5.2-1.so;osname=Linux;processor=mips64, linux/ppc64/libzstd-jni-1.5.2-1.so;osname=Linux;processor=ppc64, linux/ppc64le/libzstd-jni-1.5.2-1.so;osname=Linux;processor=ppc64le, linux/s390x/libzstd-jni-1.5.2-1.so;osname=Linux;processor=s390x, win/amd64/libzstd-jni-1.5.2-1.dll;osname=Win32;processor=amd64, win/x86/libzstd-jni-1.5.2-1.dll;osname=Win32;processor=x86

Bundle-SymbolicName: com.github.luben.zstd-jni

Bundle-Vendor: com.github.luben

Bundle-Version: 1.5.2.1

Created-By: 11.0.14 (Debian)

Export-Package: com.github.luben.zstd;version="1.5.2.1",com.github.luben.zstd.util;version="1.5.2.1"

Implementation-Title: zstd-jni

Implementation-Vendor: com.github.luben

Implementation-Vendor-Id: com.github.luben

Implementation-Version: 1.5.2-1

Import-Package: org.osgi.framework;resolution:=optional

Private-Package: linux.amd64,linux.i386,linux.aarch64,linux.arm,linux.ppc64,linux.ppc64le,linux.mips64,linux.s390x,darwin.x86_64,darwin.aar

ch64,win.amd64,win.x86,freebsd.amd64,freebsd.i386
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=11.0))
"
Specification-Title: zstd-jni
Specification-Vendor: com.github.luben
Specification-Version: 1.5.2-1
Tool: Bnd-4.0.0.201805111645

Found in path(s):

* /opt/cola/permits/1337889408_1654234488.39181/0/zstd-jni-1-5-2-1-jar/META-INF/MANIFEST.MF

1.68 okhttp 4.10.0

1.68.1 Available under license :

Note that publicsuffices.gz is compiled from The Public Suffix List:

https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:

<https://mozilla.org/MPL/2.0/>

/*

* Copyright (C) 2016 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.69 reactive-streams 1.0.4

1.69.1 Available under license :

MIT-0

1.70 url-connection-client 2.17.122

1.70.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang
Copyright 2001-2020
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed
with the accompanying software if such terms are included in the directory containing the accompanying software.
Such other license terms will then apply in lieu of the terms of the software license above.

1.71 mbknor-jackson-jsonschema_2.12 1.0.39

1.71.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 NextGenTel

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.72 kotlin-scripting-compiler-impl-embeddable 1.7.10

1.72.1 Available under license :

Apache-2.0

1.73 zstd 1.5.2

1.73.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt

otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.
BSD License

For Zstandard software

Copyright (c) 2016-present, Facebook, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name Facebook nor the names of its contributors may be used to

endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.74 protobuf-java 3.19.6

1.74.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.google.protobuf

Bnd-LastModified: 1664485905970

Build-Jdk: 1.8.0_181-google-v7

Built-By: mkruskal

Bundle-Description: Core Protocol Buffers library. Protocol Buffers are a way of encoding structured data in an efficient yet extensible format.

Bundle-DocURL: <https://developers.google.com/protocol-buffers/>

Bundle-License: <https://opensource.org/licenses/BSD-3-Clause>

Bundle-ManifestVersion: 2

Bundle-Name: Protocol Buffers [Core]

Bundle-SymbolicName: com.google.protobuf

Bundle-Version: 3.19.6

Created-By: Apache Maven Bundle Plugin

Export-Package: com.google.protobuf;version="3.19.6"

Import-Package: sun.misc;resolution:=optional,com.google.protobuf;version="[3.19,4)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"

Tool: Bnd-3.0.0.201509101326

Found in path(s):

* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.

// Redistribution and use in source and binary forms, with or without

```
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/struct.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/wrappers.proto
*
/opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/type.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/empty.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/any.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/duration.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-
jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-
jar/google/protobuf/source_context.proto
* /opt/cola/permits/1444789790_1666041710.804059/0/protobuf-java-3-19-6-jar/google/protobuf/api.proto
```

1.75 kotlin 1.6.21

1.75.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
{"version":3,"file":"kotlin.js","sources":["wrapper.js","js/arrayUtils.js","js/callableReferenceUtils.js","js/conversions.js","js/core.js","js/long.js","js/markerFunctions.js","js/misc.js","js/polyfills.js","js/rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","common/src/generated/_Arrays.kt","common/src/generated/_Ranges.kt","unsigned/src/kotlin/UByte.kt","unsigned/src/kotlin/UInt.kt","unsigned/src/kotlin/UShort.kt","builtin-sources/Ranges.kt","src/kotlin/collections/Collections.kt","src/kotlin/collections/Maps.kt","src/kotlin/collections/Sets.kt","src/kotlin/text/StringNumberConversions.kt","src/kotlin/time/Duration.kt","unsigned/src/kotlin/UnsignedUtils.kt","src/kotlin/collections/Iterables.kt","src/kotlin/collections/Sequences.kt","src/kotlin/util/Preconditions.kt","js/src/generated/_ArraysJs.kt","src/kotlin/comparisons/Comparisons.kt","src/kotlin/util/Standard.kt","js/src/generated/_ComparisonsJs.kt","unsigned/src/kotlin/ULong.kt","common/src/generated/_Collections.kt","js/src/kotlin/collections.kt","src/kotlin/collections/Iterators.kt","common/src/generated/_Comparisons.kt","common/src/generated/_Maps.kt","common/src/generated/_OneToManyTitlecaseMappings.kt","js/src/kotlin/text/char.kt","js/src/kotlin/text/string.kt","src/kotlin/text/Char.kt","src/kotlin/CharCode.kt","common/src/generated/_Sequences.kt","common/src/generated/_Sets.kt","common/src/generated/_Strings.kt","src/kotlin/text/Strings.kt","unsigned/src/kotlin/UByteArray.kt","unsigned/src/kotlin/UIntArray.kt","unsigned/src/kotlin/ULongArray.kt","unsigned/src/kotlin/UShortArray.kt","common/src/generated/_UArrays.kt","common/src/generated/_UCollections.kt","common/src/generated/_UComparisons.kt","common/src/generated/_URanges.kt","common/src/generated/_USequences.kt","common/src/kotlin/ExceptionsH.kt
```

","common/src/kotlin/JsAnnotationsH.kt","common/src/kotlin/ioH.kt","builtin-sources/Collections.kt","builtin-sources/Iterators.kt","builtin-sources/ProgressionIterators.kt","builtin-sources/Progressions.kt","builtin-sources/Range.kt","builtin-sources/Unit.kt","builtin-sources/annotation/Annotations.kt","builtin-sources/internal/InternalAnnotations.kt","builtin-sources/internal/progressionUtil.kt","src/kotlin/builtins.kt","src/kotlin/jsTypeOf.kt","src/kotlin/kotlin.kt","src/kotlin/charCode_js-v1.kt","src/kotlin/coroutines/CoroutineImpl.kt","src/kotlin/util/Result.kt","src/kotlin/coroutines/Continuation.kt","src/kotlin/coroutines/intrinsics/IntrinsicsJs.kt","src/kotlin/currentBeMisc.kt","src/kotlin/exceptions.kt","src/kotlin/jsOperators.kt","src/kotlin/math_js-v1.kt","src/kotlin/numbers_js-v1.kt","src/kotlin/reflection_js-v1.kt","src/kotlin/text/numberConversions_js-v1.kt","js/src/generated/_CharCategories.kt","js/src/generated/_CollectionsJs.kt","js/src/generated/_DigitChars.kt","js/src/generated/_LetterChars.kt","js/src/generated/_OtherLowercaseChars.kt","js/src/generated/_OtherUppercaseChars.kt","js/src/generated/_StringsJs.kt","js/src/generated/_TitlecaseMappings.kt","js/src/generated/_UArraysJs.kt","js/src/generated/_WhitespaceChars.kt","js/src/kotlin/Comparator.kt","js/src/kotlin/annotations.kt","js/src/kotlin/annotationsJVM.kt","js/src/kotlin/collections/AbstractMutableCollection.kt","js/src/kotlin/collections/AbstractMutableList.kt","js/src/kotlin/collections/AbstractMutableMap.kt","js/src/kotlin/collections/AbstractMutableSet.kt","js/src/kotlin/collections/ArrayList.kt","js/src/kotlin/collections/ArraySorting.kt","js/src/kotlin/collections/ArraysJs.kt","js/src/kotlin/collections/EqualityComparator.kt","js/src/kotlin/collections/HashMap.kt","js/src/kotlin/collections/HashSet.kt","js/src/kotlin/collections/InternalHashCodeMap.kt","js/src/kotlin/collections/InternalMap.kt","js/src/kotlin/collections/InternalStringMap.kt","js/src/kotlin/collections/LinkedHashMap.kt","js/src/kotlin/collections/LinkedHashSet.kt","js/src/kotlin/concurrent.kt","js/src/kotlin/console.kt","js/src/kotlin/coroutines/SafeContinuationJs.kt","js/src/kotlin/coroutines/cancellation/CancellationException.kt","js/src/kotlin/coroutines/js/internal/EmptyContinuation.kt","js/src/kotlin/date.kt","js/src/kotlin/dom/Builders.kt","js/src/kotlin/dom/Classes.kt","js/src/kotlin/dom/Dom.kt","js/src/kotlin/dom/EventListener.kt","js/src/kotlin/dom/ItemArrayLike.kt","js/src/kotlin/dom/Mutations.kt","js/src/kotlin/dynamic.kt","js/src/kotlin/exceptionUtils.kt","js/src/kotlin/grouping.kt","src/kotlin/collections/Grouping.kt","js/src/kotlin/internalAnnotations.kt","js/src/kotlin/json.kt","js/src/kotlin/math.kt","js/src/kotlin/numbers.kt","js/src/kotlin/promise.kt","js/src/kotlin/random/PlatformRandom.kt","js/src/kotlin/reflect/AssociatedObjects.kt","js/src/kotlin/reflect/JsClasses.kt","js/src/kotlin/reflect/KClassImpl.kt","js/src/kotlin/reflect/KClassesImpl.kt","js/src/kotlin/reflect/KTypeHelpers.kt","js/src/kotlin/reflect/KTypeImpl.kt","js/src/kotlin/reflect/KTypeParameterImpl.kt","js/src/kotlin/reflect/primitives.kt","js/src/kotlin/reflect/reflection.kt","js/src/kotlin/regexp.kt","js/src/kotlin/sequence.kt","js/src/kotlin/text/CharCategoryJS.kt","js/src/kotlin/text/CharacterCodingExceptionJs.kt","js/src/kotlin/text/StringBuilderJs.kt","js/src/kotlin/text/numberConversions.kt","js/src/kotlin/text/regex.kt","src/kotlin/text/StringBuilder.kt","js/src/kotlin/text/stringsCode.kt","js/src/kotlin/text/utf8Encoding.kt","js/src/kotlin/throwableExtensions.kt","js/src/kotlin/time/DurationJs.kt","js/src/kotlin/time/DurationUnit.kt","js/src/kotlin/time/MonoTimeSource.kt","js/src/kotlinx/dom/Builders.kt","js/src/kotlinx/dom/Classes.kt","src/kotlin/text/regex/RegexExtensions.kt","js/src/kotlinx/dom/Dom.kt","js/src/kotlinx/dom/Mutations.kt","js/src/org.w3c/deprecated.kt","js/src/org.w3c/org.khronos.webgl.kt","js/src/org.w3c/org.w3c.dom.clipboard.kt","js/src/org.w3c/org.w3c.dom.css.kt","js/src/org.w3c/org.w3c.dom.encryptedmedia.kt","js/src/org.w3c/org.w3c.dom.events.kt","js/src/org.w3c/org.w3c.dom.kt","js/src/org.w3c/org.w3c.fetch.kt","js/src/org.w3c/org.w3c.dom.mediacapture.kt","js/src/org.w3c/org.w3c.dom.mediasource.kt","js/src/org.w3c/org.w3c.dom.pointerevents.kt","js/src/org.w3c/org.w3c.dom.svg.kt","js/src/org.w3c/org.w3c.files.kt","js/src/org.w3c/org.w3c.notifications.kt","js/src/org.w3c/org.w3c.workers.kt","js/src/org.w3c/org.w3c.xhr.kt","src/kotlin/annotations/Experimental.kt","src/kotlin/annotations/ExperimentalStdlibApi.kt","src/kotlin/annotations/Inference.kt","src/kotlin/annotations/Multiplatform.kt","src/kotlin/annotations/OptIn.kt","src/kotlin/collections/AbstractCollection.kt","src/kotlin/collections/AbstractIterator.kt","src/kotlin/collections/AbstractList.kt","src/kotlin/collections/AbstractMap.kt","src/kotlin/collections/AbstractSet.kt","src/kotlin/collections/ArrayDeque.kt","src/kotlin/collections/Arrays.kt","src/kotlin/collections/BrittleContainsOptimization.kt","src/kotlin/collections/IndexedValue.kt","src/kotlin/collections/MapAccessors.kt","src/kotlin/collections/MapWithDefault.kt","src/kotlin/collections/MutableCollections.kt","src/kotlin/collections/ReversedViews.kt","src/kotlin/collections/SequenceBuilder.kt","src/kotlin/collections/SlidingWindow.kt","src/kotlin/collections/UArray

ySorting.kt","src/kotlin/comparisons/compareTo.kt","src/kotlin/contracts/ContractBuilder.kt","src/kotlin/coroutines/ContinuationInterceptor.kt","src/kotlin/coroutines/CoroutineContext.kt","src/kotlin/coroutines/CoroutineContextImpl.kt","src/kotlin/coroutines/intrinsics/Intrinsics.kt","src/kotlin/experimental/bitwiseOperations.kt","src/kotlin/experimental/inferenceMarker.kt","src/kotlin/internal/Annotations.kt","src/kotlin/properties/Delegates.kt","src/kotlin/properties/Interfaces.kt","src/kotlin/properties/ObservableProperty.kt","src/kotlin/properties/PropertyReferenceDelegates.kt","src/kotlin/random/Random.kt","src/kotlin/random/URandom.kt","src/kotlin/random/XorWowRandom.kt","src/kotlin/ranges/Ranges.kt","src/kotlin/reflect/KClasses.kt","src/kotlin/reflect/KTypeProjection.kt","src/kotlin/reflect/KVariance.kt","src/kotlin/reflect/typeOf.kt","src/kotlin/text/Appendable.kt","src/kotlin/text/Indent.kt","src/kotlin/text/Typography.kt","src/kotlin/text/regex/MatchResult.kt","src/kotlin/time/DurationUnit.kt","src/kotlin/time/ExperimentalTime.kt","src/kotlin/time/TimeSource.kt","src/kotlin/time/TimeSources.kt","src/kotlin/time/measureTime.kt","src/kotlin/util/DeepRecursive.kt","src/kotlin/util/FloorDivMod.kt","src/kotlin/util/HashCode.kt","src/kotlin/util/KotlinVersion.kt","src/kotlin/util/Lateinit.kt","src/kotlin/util/Lazy.kt","src/kotlin/util/Numbers.kt","src/kotlin/util/Suspend.kt","src/kotlin/util/Tuples.kt","unsigned/src/kotlin/UIntRange.kt","unsigned/src/kotlin/UIterators.kt","unsigned/src/kotlin/ULongRange.kt","unsigned/src/kotlin/UMath.kt","unsigned/src/kotlin/UNumbers.kt","unsigned/src/kotlin/UPrgressionUtil.kt","unsigned/src/kotlin/UStrings.kt","unsigned/src/kotlin/annotations/Unsigned.kt","common/src/kotlin/MathH.kt"],"sourcesContent":["(function

```
(root, factory) {\n  if (typeof define === 'function' && define.amd) {\n    define('kotlin', ['exports'], factory);\n  }\n  else if (typeof exports === 'object') {\n    factory(module.exports);\n  }\n  else {\n    root.kotlin = {};\n    factory(root.kotlin);\n  }\n}(this, function (Kotlin) {\n  var _ = Kotlin;\n\n  insertContent();\n});\n"/>\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
```

```
*\n\nKotlin.isArray = function (a) {\n  return (Array.isArray(a) || a instanceof Int8Array) && a.$type$ === \"BooleanArray\";\n}\n\nKotlin.isByteArray = function (a) {\n  return a instanceof Int8Array && a.$type$ !== \"BooleanArray\";\n}\n\nKotlin.isShortArray = function (a) {\n  return a instanceof Int16Array;\n}\n\nKotlin.isCharArray = function (a) {\n  return a instanceof Uint16Array && a.$type$ === \"CharArray\";\n}\n\nKotlin.isIntArray = function (a) {\n  return a instanceof Int32Array;\n}\n\nKotlin.isFloatArray = function (a) {\n  return a instanceof Float32Array;\n}\n\nKotlin.isDoubleArray = function (a) {\n  return a instanceof Float64Array;\n}\n\nKotlin.isLongArray = function (a) {\n  return Array.isArray(a) && a.$type$ === \"LongArray\";\n}\n\nKotlin.isArray = function (a) {\n  return Array.isArray(a) && !a.$type$;\n}\n\nKotlin.isArrayish = function (a) {\n  return Array.isArray(a) || ArrayBuffer.isView(a);\n}\n\nKotlin.arrayToString = function (a) {\n  if (a === null) return \"null\";\n  var toString = Kotlin.isCharArray(a) ? String.fromCharCode : Kotlin.toString;\n  return \"[\" + Array.prototype.map.call(a, function(e) { return toString(e); }).join(\", \") + \"]\";\n}\n\nKotlin.arrayDeepToString = function (arr) {\n  return Kotlin.kotlin.collections.contentDeepToStringImpl(arr);\n}\n\nKotlin.arrayEquals = function (a, b) {\n  if (a === b) {\n    return true;\n  }\n  if (a === null || b === null || !Kotlin.isArrayish(b) || a.length !== b.length) {\n    return false;\n  }\n  for (var i = 0, n = a.length; i < n; i++) {\n    if (!Kotlin.equals(a[i], b[i])) {\n      return false;\n    }\n  }\n  return true;\n}\n\nKotlin.arrayDeepEquals = function (a, b) {\n  return Kotlin.kotlin.collections.contentDeepEqualsImpl(a, b);\n}\n\nKotlin.arrayHashCode = function (arr) {\n  if (arr === null) return 0;\n  var result = 1;\n  for (var i = 0, n = arr.length; i < n; i++) {\n    result = ((31 * result | 0) + Kotlin.hashCode(arr[i])) | 0;\n  }\n  return result;\n}\n\nKotlin.arrayDeepHashCode = function (arr) {\n  return Kotlin.kotlin.collections.contentDeepHashCodeImpl(arr);\n}\n\nKotlin.primitiveArraySort = function (array) {\n  array.sort(Kotlin.doubleCompareTo);\n}\n\n"/>\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n
```

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```
*\n\nKotlin.getCallableRef = function(name, f) {\n  f.callableName = name;\n  return f;\n}\n\nKotlin.getPropertyCallableRef = function(name, paramCount, getter, setter) {\n  getter.get = getter;\n
```

```

getter.set = setter;\n  getter.callableName = name;\n  return getPropertyRefClass(getter, setter,
propertyRefClassMetadataCache[paramCount]);\n};\n\nfunction getPropertyRefClass(obj, setter, cache) {\n
obj.$metadata$ = getPropertyRefMetadata(typeof setter === \"function\" ? cache.mutable : cache.immutable);\n
obj.constructor = obj;\n  return obj;\n}\n\nvar propertyRefClassMetadataCache = [\n  {\n    mutable: { value:
null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KMutableProperty0 }\n    },\n
immutable: { value: null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KProperty0
}\n    }\n  },\n  {\n    mutable: { value: null, implementedInterface: function () {\n      return
Kotlin.kotlin.reflect.KMutableProperty1 }\n    },\n    immutable: { value: null, implementedInterface: function
() {\n      return Kotlin.kotlin.reflect.KProperty1 }\n    }\n  };\n\nfunction getPropertyRefMetadata(cache)
{\n  if (cache.value === null) {\n    cache.value = {\n      interfaces: [cache.implementedInterface()],\n
baseClass: null,\n      functions: {},\n      properties: {},\n      types: {},\n      staticMembers: {};\n
};\n  }\n  return cache.value;\n}\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.toShort = function (a) {\n  return (a & 0xFFFF) << 16 >>
16;\n};\n\nKotlin.toByte = function (a) {\n
  return (a & 0xFF) << 24 >> 24;\n};\n\nKotlin.toChar = function (a) {\n  return a &
0xFFFF;\n};\n\nKotlin.numberToLong = function (a) {\n  return a instanceof Kotlin.Long ? a :
Kotlin.Long.fromNumber(a);\n};\n\nKotlin.numberToInt = function (a) {\n  return a instanceof Kotlin.Long ?
a.toInt() : Kotlin.doubleToInt(a);\n};\n\nKotlin.numberToShort = function (a) {\n  return
Kotlin.toShort(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToByte = function (a) {\n  return
Kotlin.toByte(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToDouble = function (a) {\n  return
+a;\n};\n\nKotlin.numberToChar = function (a) {\n  return
Kotlin.toChar(Kotlin.numberToInt(a));\n};\n\nKotlin.doubleToInt = function(a) {\n  if (a > 2147483647) return
2147483647;\n  if (a < -2147483648) return -2147483648;\n  return a | 0;\n};\n\nKotlin.toBoxedChar = function
(a) {\n  if (a == null) return a;\n  if (a instanceof Kotlin.BoxedChar) return a;\n  return new
Kotlin.BoxedChar(a);\n};\n\nKotlin.unboxChar
= function(a) {\n  if (a == null) return a;\n  return Kotlin.toChar(a);\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.equals = function (obj1, obj2) {\n  if (obj1
== null) {\n    return obj2 == null;\n  }\n  if (obj2 == null) {\n    return false;\n  }\n  if (obj1 !== obj1)
{\n    return obj2 !== obj2;\n  }\n  if (typeof obj1 === \"object\" && typeof obj1.equals === \"function\") {\n
  return obj1.equals(obj2);\n  }\n  if (typeof obj1 === \"number\" && typeof obj2 === \"number\") {\n
return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2)\n  }\n  return obj1 ===
obj2;\n};\n\nKotlin.hashCode = function (obj) {\n  if (obj == null) {\n    return 0;\n  }\n  var objType = typeof
obj;\n  if (\"object\" === objType) {\n
    return \"function\" === typeof obj.hashCode ? obj.hashCode() : getObjectHashCode(obj);\n  }\n  if
(\"function\" === objType) {\n    return getObjectHashCode(obj);\n  }\n  if (\"number\" === objType) {\n
return Kotlin.numberHashCode(obj);\n  }\n  if (\"boolean\" === objType) {\n    return Number(obj)\n  }\n  var str =
String(obj);\n  return getStringHashCode(str);\n};\n\nKotlin.toString = function (o) {\n  if (o == null)
{\n    return \"null\";\n  }\n  else if (Kotlin.isArrayish(o)) {\n    return \"[...]\";\n  }\n  else {\n    return
o.toString();\n  }\n};\n\n** @const *\nvar POW_2_32 = 4294967296;\n// TODO: consider switching to Symbol
type once we are on ES6.\n** @const *\nvar OBJECT_HASH_CODE_PROPERTY_NAME =
\"kotlinHashCodeValue$\";\n\nfunction getObjectHashCode(obj) {\n  if
(! (OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {\n    var hash = (Math.random() * POW_2_32) | 0;\n    //
Make 32-bit signed integer.\n    Object.defineProperty(obj,
OBJECT_HASH_CODE_PROPERTY_NAME, { value: hash, enumerable: false });\n  }\n  return
obj[OBJECT_HASH_CODE_PROPERTY_NAME];\n}\n\nfunction getStringHashCode(str) {\n  var hash = 0;\n  for (var i = 0; i < str.length; i++) {\n    var code = str.charCodeAt(i);\n    hash = (hash * 31 + code) | 0; // Keep

```

```

it 32-bit.\n } \n return hash;\n}\n\nKotlin.identityHashCode = getObjectHashCode;\n"/\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by
the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Copyright 2009 The Closure
Library Authors. All Rights Reserved.\n/\n// Licensed under the Apache License, Version 2.0 (the "License");\n//
you may not use this file except in compliance with the License.\n// You may obtain a copy of the License at\n/\n//
http://www.apache.org/licenses/LICENSE-2.0\n/\n// Unless required by applicable law
or agreed to in writing, software\n// distributed under the License is distributed on an "AS-IS" BASIS,\n//
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n\n/**\n * Constructs a
64-bit two's-complement integer, given its low and high 32-bit\n * values as *signed* integers. See the from*
functions below for more\n * convenient ways of constructing Longs.\n */\n * The internal representation of a long is
the two given signed, 32-bit values.\n * We use 32-bit pieces because these are the size of integers on which\n *
Javascript performs bit-operations. For operations like addition and\n * multiplication, we split each number into
16-bit pieces, which can easily be\n * multiplied within Javascript's floating-point representation without overflow\n
* or change in sign.\n */\n * In the algorithms below, we frequently reduce the negative case to the\n * positive case
by negating the input(s) and then post-processing the result.\n * Note that we must ALWAYS check
specially whether those values are MIN_VALUE\n * (-2^63) because -MIN_VALUE == MIN_VALUE (since
2^63 cannot be represented as\n * a positive number, it overflows back into a negative). Not handling this\n * case
would often result in infinite recursion.\n */\n * @param {number} low The low (signed) 32 bits of the long.\n *
@param {number} high The high (signed) 32 bits of the long.\n * @constructor\n * @final\n */\nKotlin.Long =
function(low, high) {\n /**\n * @type {number}\n * @private\n */\n this.low_ = low | 0; // force into 32
signed bits.\n\n /**\n * @type {number}\n * @private\n */\n this.high_ = high | 0; // force into 32 signed
bits.\n};\n\nKotlin.Long.$metadata$ = {\n kind: "class",\n simpleName: "Long",\n interfaces: []\n};\n\n\n//
NOTE: Common constant values ZERO, ONE, NEG_ONE, etc. are defined below the\n// from* methods on which
they depend.\n\n\n/**\n * A cache of the Long representations of small integer values.\n * @type {!Object}\n *
@private\n */\nKotlin.Long.IntCache_ = {};\n\n\n/**\n * Returns a Long representing the given (32-bit) integer value.\n *
@param {number} value The 32-bit integer in question.\n * @return {!Kotlin.Long} The corresponding Long
value.\n */\nKotlin.Long.fromInt = function(value) {\n if (-128 <= value && value < 128) {\n var cachedObj =
Kotlin.Long.IntCache_[value];\n if (cachedObj) {\n return cachedObj;\n }\n }\n\n var obj = new
Kotlin.Long(value | 0, value < 0 ? -1 : 0);\n if (-128 <= value && value < 128) {\n Kotlin.Long.IntCache_[value]
= obj;\n }\n return obj;\n};\n\n\n/**\n * Converts this number value to `Long`.\n * The fractional part, if any, is
rounded down towards zero.\n * Returns zero if this `Double` value is `NaN`, `Long.MIN_VALUE` if it's less than
`Long.MIN_VALUE`,\n * `Long.MAX_VALUE` if it's bigger than `Long.MAX_VALUE`.\n * @param {number}
value The number in question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromNumber
= function(value) {\n if (isNaN(value)) {\n return Kotlin.Long.ZERO;\n } else if (value <= -
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MIN_VALUE;\n } else if (value + 1 >=
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MAX_VALUE;\n } else if (value < 0) {\n return
Kotlin.Long.fromNumber(-value).negate();\n } else {\n return new Kotlin.Long(\n (value %
Kotlin.Long.TWO_PWR_32_DBL_) | 0,\n (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0);\n
}\n};\n\n\n/**\n * Returns a Long representing the 64-bit integer that comes by concatenating\n * the given high and
low bits. Each is assumed to use 32 bits.\n * @param {number} lowBits The low 32-bits.\n * @param {number}
highBits The high 32-bits.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromBits
= function(lowBits, highBits) {\n return new Kotlin.Long(lowBits, highBits);\n};\n\n\n/**\n * Returns a Long
representation of the given string, written using the given\n *
radix.\n * @param {string} str The textual representation of the Long.\n * @param {number=} opt_radix The radix
in which the text is written.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromString
= function(str, opt_radix) {\n if (str.length == 0) {\n throw Error('number format

```



```

error: empty string');\n }\n\n var radix = opt_radix || 10;\n if (radix < 2 || 36 < radix) {\n  throw Error('radix out of
range: ' + radix);\n }\n\n if (str.charAt(0) == '-') {\n  return Kotlin.Long.fromString(str.substring(1),
radix).negate();\n } else if (str.indexOf('-') >= 0) {\n  throw Error('number format error: interior \"-\" character: ' +
str);\n }\n\n // Do several (8) digits each time through the loop, so as to\n // minimize the calls to the very
expensive emulated div.\n var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));\n var result =
Kotlin.Long.ZERO;\n for (var i = 0; i < str.length; i += 8) {\n  var size = Math.min(8, str.length
- i);\n  var value = parseInt(str.substring(i, i + size), radix);\n  if (size < 8) {\n    var power =
Kotlin.Long.fromNumber(Math.pow(radix, size));\n    result =
result.multiply(power).add(Kotlin.Long.fromNumber(value));\n  } else {\n    result =
result.multiply(radixToPower);\n    result = result.add(Kotlin.Long.fromNumber(value));\n  }\n }\n return
result;\n};\n\n\n// NOTE: the compiler should inline these constant values below and then remove\n// these
variables, so there should be no runtime penalty for these.\n\n\n/**\n * Number used repeated below in calculations.
This must appear before the\n * first call to any from* function below.\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_16_DBL_ = 1 << 16;\n\n\n/**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_24_DBL_ = 1 << 24;\n\n\n/**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_32_DBL_ =\n Kotlin.Long.TWO_PWR_16_DBL_ *
Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n
* @private\n *\nKotlin.Long.TWO_PWR_31_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ / 2;\n\n\n/**\n *
@type {number}\n * @private\n *\nKotlin.Long.TWO_PWR_48_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_
* Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_64_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_32_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_63_DBL_ =\n Kotlin.Long.TWO_PWR_64_DBL_ / 2;\n\n\n/** @type
{!Kotlin.Long} *\nKotlin.Long.ZERO = Kotlin.Long.fromInt(0);\n\n\n/** @type {!Kotlin.Long}
*\nKotlin.Long.ONE = Kotlin.Long.fromInt(1);\n\n\n/** @type {!Kotlin.Long} *\nKotlin.Long.NEG_ONE =
Kotlin.Long.fromInt(-1);\n\n\n/** @type {!Kotlin.Long} *\nKotlin.Long.MAX_VALUE =\n
Kotlin.Long.fromBits(0xFFFFFFFF | 0, 0x7FFFFFFF | 0);\n\n\n/** @type {!Kotlin.Long}
*\nKotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 | 0);\n\n\n/**\n * @type {!Kotlin.Long}\n *
@private\n *\nKotlin.Long.TWO_PWR_24_
= Kotlin.Long.fromInt(1 << 24);\n\n\n/** @return {number} The value, assuming it is a 32-bit integer.
*\nKotlin.Long.prototype.toInt = function() {\n return this.low_;\n};\n\n\n/** @return {number} The closest
floating-point representation to this value. *\nKotlin.Long.prototype.toNumber = function() {\n return this.high_ *
Kotlin.Long.TWO_PWR_32_DBL_ +\n this.getLowBitsUnsigned();\n};\n\n\n/** @return {number} The 32-bit
hashCode of this value. *\nKotlin.Long.prototype.hashCode = function() {\n return this.high_ ^
this.low_;\n};\n\n\n/**\n * @param {number=} opt_radix The radix in which the text should be written.\n * @return
{string} The textual representation of this value.\n * @override\n *\nKotlin.Long.prototype.toString =
function(opt_radix) {\n var radix = opt_radix || 10;\n if (radix < 2 || 36 < radix) {\n  throw Error('radix out of
range: ' + radix);\n }\n\n if (this.isZero()) {\n  return '0';\n }\n\n if (this.isNegative()) {\n  if
(this.equalsLong(Kotlin.Long.MIN_VALUE))\n    {\n      // We need to change the Long value before it can be negated, so we remove\n      // the bottom-most digit in
this base and then recurse to do the rest.\n      var radixLong = Kotlin.Long.fromNumber(radix);\n      var div =
this.div(radixLong);\n      var rem = div.multiply(radixLong).subtract(this);\n      return div.toString(radix) +
rem.toInt().toString(radix);\n    } else {\n      return '-' + this.negate().toString(radix);\n    }\n }\n\n // Do several (6)
digits each time through the loop, so as to\n // minimize the calls to the very expensive emulated div.\n var
radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 6));\n\n var rem = this;\n var result = '';\n while (true)
{\n  var remDiv = rem.div(radixToPower);\n  var intVal =
rem.subtract(remDiv.multiply(radixToPower)).toInt();\n  var digits = intVal.toString(radix);\n\n  rem = remDiv;\n  if (rem.isZero()) {\n    return digits + result;\n  } else {\n    while (digits.length

```

```

< 6) {\n    digits = '0' + digits;\n    }\n    result = " + digits + result;\n    }\n    };\n\n\n/** @return {number}
The high 32-bits as a signed value. */\nKotlin.Long.prototype.getHighBits = function() {\n    return
this.high_;\n};\n\n\n/** @return {number} The low 32-bits as a signed value.
*/\nKotlin.Long.prototype.getLowBits = function() {\n    return this.low_;\n};\n\n\n/** @return {number} The low
32-bits as an unsigned value. */\nKotlin.Long.prototype.getLowBitsUnsigned = function() {\n    return (this.low_ >=
0) ?\n    this.low_ : Kotlin.Long.TWO_PWR_32_DBL_ + this.low_;\n};\n\n\n\n/** @return {number} Returns
the number of bits needed to represent the absolute\n *    value of this Long.\n
*/\nKotlin.Long.prototype.getNumBitsAbs = function() {\n    if (this.isNegative()) {\n    if
(this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n    return 64;\n    } else {\n    return
this.negate().getNumBitsAbs();\n    }\n    } else {\n    var val = this.high_ != 0 ? this.high_
: this.low_;\n    for (var bit = 31; bit > 0; bit--) {\n    if ((val & (1 << bit)) != 0) {\n    break;\n    }\n    }\n    return this.high_ != 0 ? bit + 33 : bit + 1;\n    };\n};\n\n\n\n/** @return {boolean} Whether this value is zero.
*/\nKotlin.Long.prototype.isZero = function() {\n    return this.high_ == 0 && this.low_ == 0;\n};\n\n\n\n/** @return
{boolean} Whether this value is negative. */\nKotlin.Long.prototype.isNegative = function() {\n    return this.high_ <
0;\n};\n\n\n\n/** @return {boolean} Whether this value is odd. */\nKotlin.Long.prototype.isOdd = function() {\n
return (this.low_ & 1) == 1;\n};\n\n\n\n\n * @param {Kotlin.Long} other Long to compare against.\n * @return
{boolean} Whether this Long equals the other.\n */\nKotlin.Long.prototype.equalsLong = function(other) {\n    return
(this.high_ == other.high_) && (this.low_ == other.low_);\n};\n\n\n\n\n * @param {Kotlin.Long} other Long to
compare against.\n * @return {boolean} Whether this Long does not equal the other.\n
*/\nKotlin.Long.prototype.notEqualsLong = function(other) {\n    return (this.high_ != other.high_) || (this.low_ !=
other.low_);\n};\n\n\n\n\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean}
Whether this Long is less than the other.\n */\nKotlin.Long.prototype.lessThan = function(other) {\n    return
this.compare(other) < 0;\n};\n\n\n\n\n * @param {Kotlin.Long} other Long to compare against.\n * @return
{boolean} Whether this Long is less than or equal to the other.\n */\nKotlin.Long.prototype.lessThanOrEqual =
function(other) {\n    return this.compare(other) <= 0;\n};\n\n\n\n\n * @param {Kotlin.Long} other Long to
compare against.\n * @return {boolean} Whether this Long is greater than the other.\n
*/\nKotlin.Long.prototype.greaterThan = function(other) {\n    return this.compare(other) > 0;\n};\n\n\n\n\n\n *
@param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is greater than or
equal to the other.\n */\nKotlin.Long.prototype.greaterThanOrEqual
= function(other) {\n    return this.compare(other) >= 0;\n};\n\n\n\n\n\n * Compares this Long with the given one.\n *
@param {Kotlin.Long} other Long to compare against.\n * @return {number} 0 if they are the same, 1 if the this is
greater, and -1\n *    if the given one is greater.\n */\nKotlin.Long.prototype.compare = function(other) {\n    if
(this.equalsLong(other)) {\n    return 0;\n    }\n\n    var thisNeg = this.isNegative();\n    var otherNeg =
other.isNegative();\n    if (thisNeg && !otherNeg) {\n    return -1;\n    }\n    if (!thisNeg && otherNeg) {\n    return 1;\n
}\n\n    // at this point, the signs are the same, so subtraction will not overflow\n    if (this.subtract(other).isNegative())
{\n    return -1;\n    } else {\n    return 1;\n    };\n};\n\n\n\n\n\n /** @return {!Kotlin.Long} The negation of this value.
*/\nKotlin.Long.prototype.negate = function() {\n    if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n    return
Kotlin.Long.MIN_VALUE;\n    } else {\n    return this.not().add(Kotlin.Long.ONE);\n
}\n};\n\n\n\n\n\n * Returns the sum of this and the given Long.\n * @param {Kotlin.Long} other Long to add to
this one.\n * @return {!Kotlin.Long} The sum of this and the given Long.\n */\nKotlin.Long.prototype.add =
function(other) {\n    // Divide each number into 4 chunks of 16 bits, and then sum the chunks.\n\n    var a48 =
this.high_ >>> 16;\n    var a32 = this.high_ & 0xFFFF;\n    var a16 = this.low_ >>> 16;\n    var a00 = this.low_ &
0xFFFF;\n\n    var b48 = other.high_ >>> 16;\n    var b32 = other.high_ & 0xFFFF;\n    var b16 = other.low_ >>> 16;\n
var b00 = other.low_ & 0xFFFF;\n\n    var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n    c00 += a00 + b00;\n    c16 += c00
>>> 16;\n    c00 &= 0xFFFF;\n    c16 += a16 + b16;\n    c32 += c16 >>> 16;\n    c16 &= 0xFFFF;\n    c32 += a32 + b32;\n
c48 += c32 >>> 16;\n    c32 &= 0xFFFF;\n    c48 += a48 + b48;\n    c48 &= 0xFFFF;\n    return
Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n\n\n\n * Returns the difference of this and the

```

```

given Long.\n * @param {Kotlin.Long} other Long to subtract from this.\n * @return {!Kotlin.Long} The
difference of this and the given Long.\n */\nKotlin.Long.prototype.subtract = function(other) {\n return
this.add(other.negate());\n};\n\n/**\n * Returns the product of this and the given long.\n * @param {Kotlin.Long}
other Long to multiply with this.\n * @return {!Kotlin.Long} The product of this and the other.\n
*/\nKotlin.Long.prototype.multiply = function(other) {\n if (this.isZero()) {\n return Kotlin.Long.ZERO;\n } else
if (other.isZero()) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n
return other.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return this.isOdd() ? Kotlin.Long.MIN_VALUE :
Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n if (other.isNegative()) {\n return
this.negate().multiply(other.negate());\n } else {\n return this.negate().multiply(other).negate();\n
}\n } else if (other.isNegative()) {\n return this.multiply(other.negate()).negate();\n }\n\n // If both longs are
small, use float multiplication\n if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&\n
other.lessThan(Kotlin.Long.TWO_PWR_24_)) {\n return Kotlin.Long.fromNumber(this.toNumber() *
other.toNumber());\n }\n\n // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.\n // We can
skip products that would overflow.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 =
this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ &
0xFFFF;\n var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00
= 0;\n c00 += a00 * b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 * b00;\n c32 += c16 >>> 16;\n
c16 &= 0xFFFF;\n c16 += a00 * b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 +=
a32 * b00;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a16 * b16;\n c48 += c32 >>> 16;\n c32 &=
0xFFFF;\n c32 += a00 * b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 * b00 + a32 * b16 + a16 *
b32 + a00 * b48;\n c48 &= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) |
c32);\n};\n\n/**\n * Returns this Long divided by the given one.\n * @param {Kotlin.Long} other Long by which
to divide.\n * @return {!Kotlin.Long} This Long divided by the given one.\n */\nKotlin.Long.prototype.div =
function(other) {\n if (other.isZero()) {\n throw Error('division by zero');\n } else if (this.isZero()) {\n return
Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n if
(other.equalsLong(Kotlin.Long.ONE) ||\n other.equalsLong(Kotlin.Long.NEG_ONE)) {\n return
Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE == MIN_VALUE\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ONE;\n
}\n } else {\n // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.\n var halfThis =
this.shiftRight(1);\n var approx = halfThis.div(other).shiftLeft(1);\n if
(approx.equalsLong(Kotlin.Long.ZERO)) {\n return other.isNegative() ? Kotlin.Long.ONE :
Kotlin.Long.NEG_ONE;\n } else {\n var rem = this.subtract(other.multiply(approx));\n var result =
approx.add(rem.div(other));\n return result;\n }\n }\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n
if (other.isNegative()) {\n return this.negate().div(other.negate());\n } else {\n return
this.negate().div(other).negate();\n }\n } else if (other.isNegative()) {\n return
this.div(other.negate()).negate();\n }\n\n // Repeat the following until the remainder is less than other: find a\n //
floating-point that approximates remainder / other *from below*, add this\n // into
the result, and subtract it from the remainder. It is critical that\n // the approximate value is less than or equal to the
real value so that the\n // remainder never becomes negative.\n var res = Kotlin.Long.ZERO;\n var rem = this;\n
while (rem.greaterThanOrEqual(other)) {\n // Approximate the result of division. This may be a little greater or\n
// smaller than the actual value.\n var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));\n\n
// We will tweak the approximate result by changing it in the 48-th digit or\n // the smallest non-fractional digit,
whichever is larger.\n var log2 = Math.ceil(Math.log(approx) / Math.LN2);\n var delta = (log2 <= 48) ? 1 :
Math.pow(2, log2 - 48);\n\n // Decrease the approximation until it is smaller than the remainder. Note\n // that if
it is too large, the product overflows and is negative.\n var approxRes = Kotlin.Long.fromNumber(approx);\n
var approxRem = approxRes.multiply(other);\n

```

```

while (approxRem.isNegative() || approxRem.greaterThan(rem)) {
    approx -= delta;
    approxRes = Kotlin.Long.fromNumber(approx);
    approxRem = approxRes.multiply(other);
}
// We know the answer can't be zero... and actually, zero would cause
// infinite recursion since we would make no progress.
if (approxRes.isZero()) {
    approxRes = Kotlin.Long.ONE;
}
res = res.add(approxRes);
rem = rem.subtract(approxRem);
return res;
}
Returns this Long modulo the given one.
@param {Kotlin.Long} other Long by which to mod.
@return {!Kotlin.Long} This Long modulo the given one.

Kotlin.Long.prototype.modulo = function(other) {
    return this.subtract(this.div(other).multiply(other));
}
Returns the bitwise-NOT of this value.

Kotlin.Long.prototype.not = function() {
    return Kotlin.Long.fromBits(~this.low_, ~this.high_);
}
Returns the bitwise-AND of this Long and the given one.
@param {Kotlin.Long} other The Long with which to AND.
@return {!Kotlin.Long} The bitwise-AND of this and the other.

Kotlin.Long.prototype.and = function(other) {
    return Kotlin.Long.fromBits(this.low_ & other.low_,
                                this.high_ & other.high_);
}
Returns the bitwise-OR of this Long and the given one.
@param {Kotlin.Long} other The Long with which to OR.
@return {!Kotlin.Long} The bitwise-OR of this and the other.

Kotlin.Long.prototype.or = function(other) {
    return Kotlin.Long.fromBits(this.low_ | other.low_,
                                this.high_ | other.high_);
}
Returns the bitwise-XOR of this Long and the given one.
@param {Kotlin.Long} other The Long with which to XOR.
@return {!Kotlin.Long} The bitwise-XOR of this and the other.

Kotlin.Long.prototype.xor = function(other) {
    return Kotlin.Long.fromBits(this.low_ ^ other.low_,
                                this.high_ ^ other.high_);
}
Returns this Long with bits shifted to the left by the given amount.
@param {number} numBits The number of bits by which to shift.
@return {!Kotlin.Long} This shifted to the left by the given amount.

Kotlin.Long.prototype.shiftLeft = function(numBits) {
    numBits &= 63;
    if (numBits == 0) {
        return this;
    } else {
        var low = this.low_;
        if (numBits < 32) {
            var high = this.high_;
            return Kotlin.Long.fromBits(
                low << numBits, (high << numBits) | (low >>> (32 - numBits));
        ) else {
            return Kotlin.Long.fromBits(0, low << (numBits - 32));
        }
    }
}
Returns this Long with bits shifted to the right by the given amount.
@param {number} numBits The number of bits by which to shift.
@return {!Kotlin.Long} This shifted to the right by the given amount.

Kotlin.Long.prototype.shiftRight = function(numBits) {
    numBits &= 63;
    if (numBits == 0) {
        return this;
    } else {
        var high = this.high_;
        if (numBits < 32) {
            var low = this.low_;
            return Kotlin.Long.fromBits(
                (low >>> numBits) | (high << (32 - numBits)),
                high >> numBits;
        ) else {
            return Kotlin.Long.fromBits(
                high >> (numBits - 32),
                high >= 0 ? 0 : -1;
        )
    }
}
Returns this Long with bits shifted to the right by the given amount, with zeros placed into the new leading bits.
@param {number} numBits The number of bits by which to shift.
@return {!Kotlin.Long} This shifted to the right by the given amount, with zeros placed into the new leading bits.

Kotlin.Long.prototype.shiftRightUnsigned = function(numBits) {
    numBits &= 63;
    if (numBits == 0) {
        return this;
    } else {
        var high = this.high_;
        if (numBits < 32) {
            var low = this.low_;
            return Kotlin.Long.fromBits(
                (low >>> numBits) | (high << (32 - numBits)),
                high >>> numBits;
        ) else if (numBits == 32) {
            return Kotlin.Long.fromBits(high, 0);
        } else {
            return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);
        }
    }
}
Support for Kotlin

Kotlin.Long.prototype.equals = function (other) {
    return other instanceof Kotlin.Long && this.equalsLong(other);
}
Kotlin.Long.prototype.compareTo = function (other) {
    return this.compareToLong(other);
}
Kotlin.Long.prototype.compare = function (other) {
    return this.compareToLong(other);
}
Kotlin.Long.prototype.inc = function() {
    return this.add(Kotlin.Long.ONE);
}
Kotlin.Long.prototype.dec = function() {
    return this.add(Kotlin.Long.NEG_ONE);
}
Kotlin.Long.prototype.valueOf = function() {
    return this.toNumber();
}
Kotlin.Long.prototype.unaryPlus = function() {
    return this;
}
Kotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;
Kotlin.Long.prototype.inv = Kotlin.Long.prototype.not;
Kotlin.Long.prototype.rangeTo = function (other) {
    return new Kotlin.Long.rangeToLong(this, other);
}

```

```

Kotlin.kotlin.ranges.LongRange(this,
other);\n};\n";\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n/**\n * @param {string} id\n * @param {Object} declaration\n */\nKotlin.defineModule = function (id,
declaration) {\n};\n\nKotlin.defineInlineFunction = function (tag, fun) {\n return fun;\n};\n\nKotlin.wrapFunction
= function (fun) {\n var f = function () {\n f = fun();\n return f.apply(this, arguments);\n }; \n return
function () {\n return f.apply(this, arguments);\n }; \n};\n\nKotlin.isTypeOf = function (type) {\n return
function (object) {\n return typeof object === type;\n }; \n};\n\nKotlin.isInstanceOf = function (klass) {\n
return function (object) {\n return Kotlin.isType(object, klass);\n }; \n};\n\nKotlin.orNull = function (fn) {\n
return function (object) {\n return object == null || fn(object);\n
}; \n};\n\nKotlin.andPredicate = function (a, b) {\n return function (object) {\n return a(object) &&
b(object);\n }; \n};\n\nKotlin.kotlInModuleMetadata = function (abiVersion, moduleName, data)
{\n};\n\nKotlin.suspendCall = function (value) {\n return value;\n};\n\nKotlin.coroutineResult = function (qualifier)
{\n throwMarkerError();\n};\n\nKotlin.coroutineController = function (qualifier) {\n
throwMarkerError();\n};\n\nKotlin.coroutineReceiver = function (qualifier) {\n
throwMarkerError();\n};\n\nKotlin.setCoroutineResult = function (value, qualifier) {\n
throwMarkerError();\n};\n\nKotlin.getReifiedTypeParameterKType = function (typeParameter) {\n
throwMarkerError();\n};\n\nfunction throwMarkerError() {\n throw new Error(\n "\nThis marker function
should never be called. \n" +\n "\nLooks like compiler did not eliminate it properly. \n" +\n "\nPlease, report
an issue if you caught this exception.\n");\n}\n\nKotlin.getFunctionById
= function (id, defaultValue) {\n return function () {\n return defaultValue;\n }; \n};\n";\n/*\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by
the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.compareTo = function (a, b)
{\n var typeA = typeof a;\n if (typeA === "number") {\n if (typeof b === "number") {\n return
Kotlin.doubleCompareTo(a, b);\n } \n return Kotlin.primitiveCompareTo(a, b);\n } \n if (typeA ===
"string" || typeA === "boolean") {\n return Kotlin.primitiveCompareTo(a, b);\n } \n return
a.compareTo_11rb$(b);\n};\n\nKotlin.primitiveCompareTo = function (a, b) {\n return a < b ? -1 : a > b ? 1 :
0;\n};\n\nKotlin.doubleCompareTo = function (a, b) {\n if (a < b) return -1;\n if (a > b) return 1;\n\n if (a ===
b) {\n if (a !== 0) return 0;\n\n var ia = 1 / a;\n
return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n } \n\n return a !== a ? (b !== b ? 0 : 1) : -1;\n};\n\nKotlin.charInc =
function (value) {\n return Kotlin.toChar(value+1);\n};\n\nKotlin.charDec = function (value) {\n return
Kotlin.toChar(value-1);\n};\n\nKotlin.imul = Math.imul || imul;\n\nKotlin.imulEmulated = imul;\n\nfunction imul(a,
b) {\n return ((a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;\n}\n\n(function () {\n var buf = new
ArrayBuffer(8);\n var bufFloat64 = new Float64Array(buf);\n var bufFloat32 = new Float32Array(buf);\n var
bufInt32 = new Int32Array(buf);\n var lowIndex = 0;\n var highIndex = 1;\n\n bufFloat64[0] = -1; //
bff00000_00000000\n if (bufInt32[lowIndex] !== 0) {\n lowIndex = 1;\n highIndex = 0;\n } \n\n
Kotlin.doubleToBits = function (value) {\n return Kotlin.doubleToRawBits(isNaN(value) ? NaN : value);\n
};\n\n Kotlin.doubleToRawBits = function (value) {\n bufFloat64[0]
= value;\n return Kotlin.Long.fromBits(bufInt32[lowIndex], bufInt32[highIndex]);\n }; \n\n
Kotlin.doubleFromBits = function (value) {\n bufInt32[lowIndex] = value.low_;\n bufInt32[highIndex] =
value.high_;\n return bufFloat64[0];\n }; \n\n Kotlin.floatToBits = function (value) {\n return
Kotlin.floatToRawBits(isNaN(value) ? NaN : value);\n }; \n\n Kotlin.floatToRawBits = function (value) {\n
bufFloat32[0] = value;\n return bufInt32[0];\n }; \n\n Kotlin.floatFromBits = function (value) {\n
bufInt32[0] = value;\n return bufFloat32[0];\n }; \n\n // returns zero value for number with positive sign bit
and non-zero value for number with negative sign bit.\n Kotlin.doubleSignBit = function (value) {\n
bufFloat64[0] = value;\n return bufInt32[highIndex] & 0x80000000;\n }; \n\n Kotlin.numberHashCode =
function (obj) {\n if ((obj | 0) === obj) {\n return obj | 0;\n

```

```

    }\n    else {\n        bufFloat64[0] = obj;\n        return (bufInt32[highIndex] * 31 | 0) + bufInt32[lowIndex] |
0;\n    }\n    }\n});\n\nKotlin.ensureNotNull = function(x) {\n    return x != null ? x :
Kotlin.throwNPE();\n};\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nif (typeof String.prototype.startsWith === "undefined") {\n
Object.defineProperty(String.prototype, "startsWith", {\n    value: function (searchString, position) {\n
position = position || 0;\n        return this.lastIndexOf(searchString, position) === position;\n    }\n});\n\nif (typeof String.prototype.endsWith === "undefined") {\n    Object.defineProperty(String.prototype, "endsWith",
{\n    value: function (searchString, position) {\n        var subjectString = this.toString();\n
        if (position === undefined || position > subjectString.length) {\n            position = subjectString.length;\n
        }\n        position -= searchString.length;\n        var lastIndex = subjectString.indexOf(searchString,
position);\n        return lastIndex !== -1 && lastIndex === position;\n    }\n});\n\n// ES6 Math polyfills\nif (typeof Math.sign === "undefined") {\n    Math.sign = function(x) {\n        x = +x; // convert to a number\n
        if (x === 0 || isNaN(x)) {\n            return Number(x);\n        }\n        return x > 0 ? 1 : -1;\n    };}\n\nif (typeof
Math.trunc === "undefined") {\n    Math.trunc = function(x) {\n        if (isNaN(x)) {\n            return NaN;\n
        }\n        if (x > 0) {\n            return Math.floor(x);\n        }\n        return Math.ceil(x);\n    };}\n\n(function() {\n
var epsilon = 2.220446049250313E-16;\n    var taylor_2_bound = Math.sqrt(epsilon);\n    var taylor_n_bound =
Math.sqrt(taylor_2_bound);\n
    var upper_taylor_2_bound = 1/taylor_2_bound;\n    var upper_taylor_n_bound = 1/taylor_n_bound;\n\n    if
(typeof Math.sinh === "undefined") {\n        Math.sinh = function(x) {\n            if (Math.abs(x) < taylor_n_bound)
{\n                var result = x;\n                if (Math.abs(x) > taylor_2_bound) {\n                    result += (x * x * x) / 6;\n
                }\n                return result;\n            } else {\n                var y = Math.exp(x);\n                var y1 = 1 / y;\n
                if (!isFinite(y)) return Math.exp(x - Math.LN2);\n                if (!isFinite(y1)) return -Math.exp(-x - Math.LN2);\n
                return (y - y1) / 2;\n            }\n        };}\n\n    if (typeof Math.cosh === "undefined") {\n        Math.cosh
= function(x) {\n        var y = Math.exp(x);\n        var y1 = 1 / y;\n        if (!isFinite(y) || !isFinite(y1)) return
Math.exp(Math.abs(x) - Math.LN2);\n        return (y + y1) / 2;\n    };}\n\n    if (typeof Math.tanh === "undefined") {\n        Math.tanh = function(x){\n            if
(Math.abs(x) < taylor_n_bound) {\n                var result = x;\n                if (Math.abs(x) > taylor_2_bound) {\n
                    result -= (x * x * x) / 3;\n                }\n                return result;\n            } else {\n                var a =
Math.exp(+x), b = Math.exp(-x);\n                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);\n
            }\n        };}\n\n    // Inverse hyperbolic function implementations derived from boost special math functions,\n
// Copyright Eric Ford & Hubert Holin 2001.\n\n    if (typeof Math.asinh === "undefined") {\n        var asinh =
function(x) {\n            if (x >= +taylor_n_bound)\n                {\n                    if (x > upper_taylor_n_bound)\n
                    {\n                        //
                    approximation by laurent series in 1/x at 0+ order from -1 to 0\n                        return Math.log(x) + Math.LN2;\n
                    }\n                    else\n                        {\n                            // approximation by laurent series in 1/x at 0+ order
from -1 to 1\n                            return Math.log(x * 2 + (1 / (x * 2)));\n                        }\n                    }\n                    else\n                        {\n                            return Math.log(x + Math.sqrt(x * x + 1));\n
                        }\n                    }\n                } else if (x <= -
taylor_n_bound)\n                {\n                    return -asinh(-x);\n                }\n                else\n                {\n                    //
                    approximation by taylor series in x at 0 up to order 2\n                    var result = x;\n                    if (Math.abs(x) >=
taylor_2_bound)\n                        {\n                            var x3 = x * x * x;\n                            // approximation by taylor series in x at
0 up to order 4\n                            result
                            -= x3 / 6;\n                        }\n                    }\n                }\n                return result;\n            }\n        };}\n\n    Math.asinh = asinh;\n\n    if (typeof
Math.acosh === "undefined") {\n        Math.acosh = function(x) {\n            if (x < 1)\n                {\n                    return
NaN;\n                }\n                else if (x - 1 >= taylor_n_bound)\n                {\n                    if (x > upper_taylor_2_bound)\n
                    {\n                        //
                    approximation by laurent series in 1/x at 0+ order from -1 to 0\n                        return
Math.log(x) + Math.LN2;\n                    }\n                    else\n                        {\n                            return Math.log(x + Math.sqrt(x *
x - 1));\n                        }\n                    }\n                }\n                else\n                {\n                    var y = Math.sqrt(x - 1);\n                    //

```



```

Object.defineProperty(Function.prototype, 'apply', {\n      value: function(self, array) {\n          return
apply.call(this, self, [].slice.call(array));\n      }\n  });\n  }\n\n  // Patch map to work with TypedArrays if
needed.\n  for (var i = 0; i < arrays.length; ++i) {\n      var TypedArray = arrays[i];\n      if (typeof
TypedArray.prototype.map === \"undefined\") {\n          Object.defineProperty(TypedArray.prototype, 'map', {\n
              value: function(callback, self) {\n
                  return [].slice.call(this).map(callback, self);\n              }\n          });\n      }\n  }\n\n  // Patch sort to work
with TypedArrays if needed.\n  // TODO: consider to remove following function and replace it with
`Kotlin.doubleCompareTo` (see misc.js)\n  var totalOrderComparator = function (a, b) {\n      if (a < b) return -
1;\n      if (a > b) return 1;\n      if (a === b) {\n          if (a !== 0) return 0;\n          var ia = 1 / a;\n
return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n      }\n      return a !== a ? (b !== b ? 0 : 1) : -1\n  };\n\n  for (var i =
0; i < arrays.length; ++i) {\n      var TypedArray = arrays[i];\n      if (typeof TypedArray.prototype.sort ===
\"undefined\") {\n          Object.defineProperty(TypedArray.prototype, 'sort', {\n              value:
function(compareFunction) {\n                  return Array.prototype.sort.call(this, compareFunction ||
totalOrderComparator);\n              }\n          });\n      }\n  }\n\n  };\n  }\n\n  }());\n  \"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.Kind = {\n  CLASS: \"class\",\n  INTERFACE: \"interface\",\n  OBJECT: \"object\"};\n\nKotlin.callGetter = function (thisObject, klass, propertyName) {\n  var
propertyDescriptor = Object.getOwnPropertyDescriptor(klass, propertyName);\n  if (propertyDescriptor != null
&& propertyDescriptor.get != null) {\n      return propertyDescriptor.get.call(thisObject);\n  }\n\n  propertyDescriptor = Object.getOwnPropertyDescriptor(thisObject, propertyName);\n  if (propertyDescriptor !=
null && \"value\" in propertyDescriptor) {\n      return thisObject[propertyName];\n  }\n\n  return
Kotlin.callGetter(thisObject, Object.getPrototypeOf(klass), propertyName);\n};\n\nKotlin.callSetter = function
(thisObject, klass,
  propertyName, value) {\n  var propertyDescriptor = Object.getOwnPropertyDescriptor(klass, propertyName);\n
if (propertyDescriptor != null && propertyDescriptor.set != null) {\n      propertyDescriptor.set.call(thisObject,
value);\n      return;\n  }\n\n  propertyDescriptor = Object.getOwnPropertyDescriptor(thisObject,
propertyName);\n  if (propertyDescriptor != null && \"value\" in propertyDescriptor) {\n      thisObject[propertyName] = value;\n      return\n  }\n\n  Kotlin.callSetter(thisObject,
Object.getPrototypeOf(klass), propertyName, value);\n};\n\nfunction isInheritanceFromInterface(ctor, iface) {\n  if
(ctor === iface) return true;\n  var metadata = ctor.$metadata$;\n  if (metadata != null) {\n      var interfaces =
metadata.interfaces;\n      for (var i = 0; i < interfaces.length; i++) {\n          if
(isInheritanceFromInterface(interfaces[i], iface)) {\n              return true;\n          }\n      }\n  }\n\n  var
superPrototype
= ctor.prototype != null ? Object.getPrototypeOf(ctor.prototype) : null;\n  var superConstructor = superPrototype
!= null ? superPrototype.constructor : null;\n  return superConstructor != null &&
isInheritanceFromInterface(superConstructor, iface);\n}\n\n**\n * @param {*} object\n * @param
{Function|Object} klass\n * @returns {Boolean}\n */\n\nKotlin.isType = function (object, klass) {\n  if (klass ===
Object) {\n      switch (typeof object) {\n          case \"string\":\n          case \"number\":\n          case
\"boolean\":\n          case \"function\":\n              return true;\n          default:\n              return object instanceof
Object;\n      }\n  }\n\n  if (object == null || klass == null || (typeof object !== 'object' && typeof object !==
'function')) {\n      return false;\n  }\n\n  if (typeof klass === \"function\" && object instanceof klass) {\n
return true;\n  }\n\n  var proto = Object.getPrototypeOf(klass);\n
var constructor = proto != null ? proto.constructor : null;\n  if (constructor != null && \"$metadata$\" in
constructor) {\n      var metadata = constructor.$metadata$;\n      if (metadata.kind === Kotlin.Kind.OBJECT) {\n
return object === klass;\n      }\n  }\n\n  var classMetadata = klass.$metadata$;\n  // In WebKit
(JavaScriptCore) for some interfaces from DOM typeof returns \"object\", nevertheless they can be used in RHS of
instanceof\n  if (classMetadata == null) {\n      return object instanceof klass;\n  }\n\n  if (classMetadata.kind

```



```

=== Kotlin.Kind.INTERFACE && object.constructor != null) {\n    return
isInheritanceFromInterface(object.constructor, class);\n }\n\n return false;\n};\n\nKotlin.isNumber = function (a)
{\n    return typeof a === \"number\" || a instanceof Kotlin.Long;\n};\n\nKotlin.isChar = function (value) {\n    return
value instanceof Kotlin.BoxedChar;\n};\n\nKotlin.isComparable = function (value) {\n    var type = typeof
value;\n\n    return type === \"string\" ||\n        type === \"boolean\" ||\n        Kotlin.isNumber(value) ||\n        Kotlin.isType(value, Kotlin.kotlin.Comparable);\n};\n\nKotlin.isCharSequence = function (value) {\n    return
typeof value === \"string\" || Kotlin.isType(value, Kotlin.kotlin.CharSequence);\n};\n\n/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get
declarations directly under the module\n\n@PublishedApi\n\ninternal fun <T> Array(size: Int):
Array<T>\n\n@JsName(\"newArray\")\n\nfun <T> newArray(size: Int, initValue: T) = fillArrayVal(Array<T>(size),
initValue)\n\n@JsName(\"newArrayF\")\n\ninline fun <T> arrayWithFun(size: Int, init: (Int) -> T) =
fillArrayFun(Array<T>(size), init)\n\n@JsName(\"fillArray\")\n\ninline fun <T> fillArrayFun(array: Array<T>, init:
(Int)
-> T): Array<T> {\n    for (i in 0..array.size - 1) {\n        array[i] = init(i)\n    }\n    return
array\n}\n\n@JsName(\"booleanArray\")\n\nfun booleanArray(size: Int, init: dynamic): Array<Boolean> {\n    val
result: dynamic = Array<Boolean>(size)\n    result.`$type$` = \"BooleanArray\"\n    return when (init) {\n        null,
true -> fillArrayVal(result, false)\n        false -> result\n        else -> fillArrayFun<Boolean>(result, init)\n
}\n}\n\n@JsName(\"booleanArrayF\")\n\ninline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean):
Array<Boolean> = fillArrayFun(booleanArray(size, false),
init)\n\n@JsName(\"charArray\")\n\n@Suppress(\"UNUSED_PARAMETER\")\n\nfun charArray(size: Int, init:
dynamic): Array<Char> {\n    val result = js(\"new Uint16Array(size)\")\n    result.`$type$` = \"CharArray\"\n    return when (init) {\n        null, true, false -> result // For consistency\n        else -> fillArrayFun<Char>(result,
init)\n    }\n}\n\n@JsName(\"charArrayF\")\n\ninline fun charArrayWithFun(size:
Int, init: (Int) -> Char): Array<Char> {\n    val array = charArray(size, null)\n    for (i in 0..array.size - 1) {\n
@Suppress(\"UNUSED_VARIABLE\") // used in js block\n        val value = init(i)\n        js(\"array[i] = value;\")\n
}\n    return array\n}\n\n@JsName(\"untypedCharArrayF\")\n\ninline fun untypedCharArrayWithFun(size: Int, init:
(Int) -> Char): Array<Char> {\n    val array = Array<Char>(size)\n    for (i in 0..array.size - 1) {\n
@Suppress(\"UNUSED_VARIABLE\") // used in js block\n        val value = init(i)\n        js(\"array[i] = value;\")\n
}\n    return array\n}\n\n@JsName(\"longArray\")\n\nfun longArray(size: Int, init: dynamic): Array<Long> {\n    val
result: dynamic = Array<Long>(size)\n    result.`$type$` = \"LongArray\"\n    return when (init) {\n        null, true ->
fillArrayVal(result, 0L)\n        false -> result\n        else -> fillArrayFun<Long>(result, init)\n
}\n}\n\n@JsName(\"longArrayF\")\n\ninline fun longArrayWithFun(size:
Int, init: (Int) -> Long): Array<Long> = fillArrayFun(longArray(size, false), init)\n\nprivate fun <T>
fillArrayVal(array: Array<T>, initValue: T): Array<T> {\n    for (i in 0..array.size - 1) {\n        array[i] = initValue\n
}\n    return array\n}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin\n\npublic class Enum<T> : Comparable<Enum<T>>
{\n    @JsName(\"name$\") private var _name: String = \"\"\n    @JsName(\"ordinal$\") private var _ordinal: Int =
0\n\n    val name: String\n        get() = _name\n\n    val ordinal: Int\n        get() = _ordinal\n\n    override fun
compareTo(other: Enum<T>) = ordinal.compareTo(other.ordinal)\n\n    override fun equals(other: Any?) = this ===
other\n\n    override fun hashCode(): Int = js(\"Kotlin.identityHashCode\")(this)\n\n    override fun
toString() = name\n\n    companion object\n}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js.internal\n\n@JsName(\"DoubleCompanionObject\")\n\ninternal object DoubleCompanionObject {\n
\n    @JsName(\"MIN_VALUE\")\n    const val MIN_VALUE: Double = 4.9E-324\n\n    @JsName(\"MAX_VALUE\")\n    const val MAX_VALUE: Double = 1.7976931348623157E308\n\n}

```

```

@JsName("POSITIVE_INFINITY")\n @Suppress("DIVISION_BY_ZERO")\n const val
POSITIVE_INFINITY: Double = 1.0 / 0.0\n\n @JsName("NEGATIVE_INFINITY")\n
@Suppress("DIVISION_BY_ZERO")\n const val NEGATIVE_INFINITY: Double = -1.0 / 0.0\n\n
@JsName("NaN")\n @Suppress("DIVISION_BY_ZERO")\n const val NaN: Double = -(0.0 / 0.0)\n\n
@JsName("SIZE_BYTES")\n const val SIZE_BYTES = 8\n\n @JsName("SIZE_BITS")\n const val
SIZE_BITS
= 64\n}\n\n@JsName("FloatCompanionObject")\ninternal object FloatCompanionObject {\n
@JsName("MIN_VALUE")\n const val MIN_VALUE: Float = 1.4E-45F\n\n @JsName("MAX_VALUE")\n
const val MAX_VALUE: Float = 3.4028235E38F\n\n @JsName("POSITIVE_INFINITY")\n
@Suppress("DIVISION_BY_ZERO")\n const val POSITIVE_INFINITY: Float = 1.0F / 0.0F\n\n
@JsName("NEGATIVE_INFINITY")\n @Suppress("DIVISION_BY_ZERO")\n const val
NEGATIVE_INFINITY: Float = -1.0F / 0.0F\n\n @JsName("NaN")\n
@Suppress("DIVISION_BY_ZERO")\n const val NaN: Float = -(0.0F / 0.0F)\n\n
@JsName("SIZE_BYTES")\n const val SIZE_BYTES = 4\n\n @JsName("SIZE_BITS")\n const val
SIZE_BITS = 32\n}\n\n@JsName("IntCompanionObject")\ninternal object IntCompanionObject {\n
@JsName("MIN_VALUE")\n val MIN_VALUE: Int = -2147483647 - 1\n\n @JsName("MAX_VALUE")\n
val MAX_VALUE: Int = 2147483647\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 4\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 32\n}\n\n@JsName("LongCompanionObject")\ninternal
object LongCompanionObject {\n @JsName("MIN_VALUE")\n val MIN_VALUE: Long =
js("Kotlin.Long.MIN_VALUE")\n\n @JsName("MAX_VALUE")\n val MAX_VALUE: Long =
js("Kotlin.Long.MAX_VALUE")\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 8\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 64\n}\n\n@JsName("ShortCompanionObject")\ninternal
object ShortCompanionObject {\n @JsName("MIN_VALUE")\n val MIN_VALUE: Short = -32768\n\n
@JsName("MAX_VALUE")\n val MAX_VALUE: Short = 32767\n\n @JsName("SIZE_BYTES")\n const
val SIZE_BYTES = 2\n\n @JsName("SIZE_BITS")\n const val SIZE_BITS =
16\n}\n\n@JsName("ByteCompanionObject")\ninternal object ByteCompanionObject {\n
@JsName("MIN_VALUE")\n val MIN_VALUE: Byte = -128\n\n @JsName("MAX_VALUE")\n val
MAX_VALUE: Byte = 127\n\n @JsName("SIZE_BYTES")\n const
val SIZE_BYTES = 1\n\n @JsName("SIZE_BITS")\n const val SIZE_BITS =
8\n}\n\n@JsName("CharCompanionObject")\ninternal object CharCompanionObject {\n
@JsName("MIN_VALUE")\n public const val MIN_VALUE: Char = "\u0000"\n\n
@JsName("MAX_VALUE")\n public const val MAX_VALUE: Char = "\uFFFF"\n\n
@JsName("MIN_HIGH_SURROGATE")\n public const val MIN_HIGH_SURROGATE: Char = "\uD800"\n\n
@JsName("MAX_HIGH_SURROGATE")\n public const val MAX_HIGH_SURROGATE: Char =
"\uDBFF"\n\n @JsName("MIN_LOW_SURROGATE")\n public const val MIN_LOW_SURROGATE: Char =
"\uDC00"\n\n @JsName("MAX_LOW_SURROGATE")\n public const val MAX_LOW_SURROGATE: Char
= "\uDFFF"\n\n @JsName("MIN_SURROGATE")\n public const val MIN_SURROGATE: Char =
MIN_HIGH_SURROGATE\n\n @JsName("MAX_SURROGATE")\n public const val MAX_SURROGATE:
Char = MAX_LOW_SURROGATE\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 2\n\n
@JsName("SIZE_BITS")\n const
val SIZE_BITS = 16\n}\n\ninternal object StringCompanionObject {} \n\ninternal object
BooleanCompanionObject {} \n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the

```

size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component1(): T
{\n  return get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component1(): Byte {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component1(): Short {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component1(): Int {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component1(): Long {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component1(): Float {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component1(): Double {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component1(): Boolean {\n  return
get(0)\n}\n\n**\n

```

* Returns 1st *element* from the array.

* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component1(): Char {\n  return
get(0)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component2(): T {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component2(): Byte {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component2(): Short {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component2(): Int {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component2(): Long {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component2(): Float {\n  return
get(1)\n}\n\n**\n

```

* Returns 2nd *element* from the array.

* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*^@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component2():

```

```

Double {\n    return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is
less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component2(): Boolean {\n
    return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2,
throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component2(): Char {\n    return
get(1)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component3(): T {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component3(): Byte {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n
* \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where
the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component3(): Short {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less
than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component3(): Int {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component3(): Long {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component3(): Float {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component3(): Double {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component3(): Boolean {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n
* where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun
CharArray.component3(): Char {\n    return get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If
the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out
T>.component4(): T {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of
this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is
unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component4(): Byte {\n
    return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4,
throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic
inline operator fun ShortArray.component4(): Short {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from
the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in
Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun
IntArray.component4(): Int {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If
the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component4(): Long {\n

```

return get(3)}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component4(): Float {\n return get(3)}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component4(): Double {\n return get(3)}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component4(): Boolean {\n return get(3)}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component4(): Char {\n return get(3)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component5(): T {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component5(): Byte {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component5(): Short {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component5(): Int {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component5(): Long {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component5(): Float {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component5(): Double {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component5(): Boolean {\n return get(4)}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component5(): Char {\n return get(4)}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {\n return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun ByteArray.contains(element: Byte): Boolean {\n return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun ShortArray.contains(element: Short): Boolean {\n return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun IntArray.contains(element: Int): Boolean {\n return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun LongArray.contains(element: Long): Boolean {\n return

```

indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n@Deprecated("The
function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it ==
element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior
as in a list.", ReplaceWith("any { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.6")\n@Suppress("DEPRECATION_ERROR")\npublic operator fun FloatArray.contains(element:
Float): Boolean {\n    return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n
*/\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed
soon. Use 'any { it == element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the
same search behavior as in a list.", ReplaceWith("any { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6")\n@Suppress("DEPRECATION_ERROR")\npublic operator fun DoubleArray.contains(element: Double):
Boolean {\n    return indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n
*/\n\npublic operator fun BooleanArray.contains(element: Boolean): Boolean {\n    return
indexOf(element) >= 0}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\npublic operator fun
CharArray.contains(element: Char): Boolean {\n    return indexOf(element) >= 0}\n\n/**\n * Returns an element
at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n *
@sample samples.collections.Collections.Elements.elementAt\n */\n\npublic expect fun <T> Array<out
T>.elementAt(index: Int): T\n\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic expect fun ByteArray.elementAt(index: Int):
Byte\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic
expect fun ShortArray.elementAt(index:
Int): Short\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n\npublic expect fun IntArray.elementAt(index: Int): Int\n\n/**\n * Returns an element at the given [index] or
throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic expect fun LongArray.elementAt(index: Int):
Long\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic
expect fun FloatArray.elementAt(index: Int): Float\n\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n
*/\n\npublic expect fun DoubleArray.elementAt(index: Int): Double\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic expect fun BooleanArray.elementAt(index: Int):
Boolean\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n\npublic expect fun CharArray.elementAt(index: Int): Char\n\n/**\n * Returns an element at the given [index] or
the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return
if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the
given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n *
@sample samples.collections.Collections.Elements.elementAtOrElse\n */\n\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample

```

```

samples.collections.Collections.Elements.elementAtOrElse\n *^n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {\n  return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at
the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n *
\n * @sample samples.collections.Collections.Elements.elementAtOrElse\n *^n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {\n  return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *^n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Long): Long {\n  return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*^n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.elementAtOrElse(index: Int, defaultValue: (Int) ->
Float): Float {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n *
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*^n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.elementAtOrElse(index: Int, defaultValue: (Int) ->
Double): Double {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n
* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*^n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n  return if
(index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the
given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n *
\n * @sample samples.collections.Collections.Elements.elementAtOrElse\n *^n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n  return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or
`null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *^n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrNull(index: Int): T? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given
[index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *^n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.elementAtOrNull(index: Int): Byte? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *^n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.elementAtOrNull(index: Int): Short? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *^n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.elementAtOrNull(index: Int): Int? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n
*^n@kotlin.internal.InlineOnly\npublic inline fun LongArray.elementAtOrNull(index: Int): Long? {\n  return
this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds
of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n
*^n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.elementAtOrNull(index: Int): Float? {\n  return
this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds

```

```

of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n
*\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.elementAtOrNull(index: Int): Double? {\n  return
this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds
of this array.\n * \n * @sample
  samples.collections.Collections.Elements.elementAtOrNull\n *\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.elementAtOrNull(index: Int): Boolean? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
  samples.collections.Collections.Elements.elementAtOrNull\n *\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.elementAtOrNull(index: Int): Char? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns the first
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.find(predicate: (T) -> Boolean): T? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n
  *\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? {\n  return
firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
  *\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.find(predicate: (Short) -> Boolean): Short? {\n
  return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
  *\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.find(predicate: (Int) -> Boolean): Int? {\n  return
firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
  *\n@kotlin.internal.InlineOnly\npublic
  inline fun LongArray.find(predicate: (Long) -> Boolean): Long? {\n  return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.find(predicate: (Float) -> Boolean): Float? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.find(predicate: (Double) -> Boolean): Double? {\n  return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.find(predicate:
  (Boolean) -> Boolean): Boolean? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.find(predicate: (Char) -> Boolean): Char? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.findLast(predicate: (T) -> Boolean): T? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
  samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {\n
  return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
  *\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {\n
  return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no

```



```

such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
*\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {\n  return
lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
*\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {\n
return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.findLast(predicate: (Float) -> Boolean): Float? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {\n  return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {\n  return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.findLast(predicate: (Char) -> Boolean): Char? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns
first element.\n * @throws [NoSuchElementException] if the array is empty.\n *\npublic fun <T> Array<out
T>.first(): T {\n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n
*\npublic fun ByteArray.first(): Byte {\n  if (isEmpty())\n    throw NoSuchElementException("Array is
empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the
array is empty.\n *\npublic fun ShortArray.first(): Short {\n  if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n
* Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n *\npublic fun
IntArray.first(): Int {\n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n
*\npublic fun LongArray.first(): Long {\n  if (isEmpty())\n    throw NoSuchElementException("Array is
empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the
array is empty.\n *\npublic fun FloatArray.first(): Float {\n  if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws
[NoSuchElementException] if the array is empty.\n *\npublic fun DoubleArray.first(): Double {\n  if
(isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns first
element.\n
* @throws [NoSuchElementException] if the array is empty.\n *\npublic fun BooleanArray.first(): Boolean {\n
if (isEmpty())\n  throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns
first element.\n * @throws [NoSuchElementException] if the array is empty.\n *\npublic fun CharArray.first():
Char {\n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n
* Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n *\npublic inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {\n  for (element in
this) if (predicate(element)) return element\n  throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\npublic inline fun ByteArray.first(predicate:
(Byte) -> Boolean): Byte {\n  for (element in this) if (predicate(element)) return element\n  throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n

```

```

*^public inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun IntArray.first(predicate: (Int) -> Boolean): Int {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun LongArray.first(predicate: (Long) -> Boolean): Long {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first element matching the given [predicate].
// @throws [NoSuchElementException] if no such element is found.
*^public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
// Returns the first non-null value produced by [transform] function being applied to elements of this array in iteration order.
// @throws [NoSuchElementException] if no non-null value was produced.
// @sample samples.collections.Collections.Transformations.firstNotNullOf
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly
public inline fun <T, R : Any> Array<out T>.firstNotNullOf(transform: (T) -> R?): R {
    return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException("No element of the array was transformed to a non-null value.")
}
// Returns the first non-null value produced by [transform] function being applied to elements of this array in iteration order.
// @sample samples.collections.Collections.Transformations.firstNotNullOfOrNull
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly
public inline fun <T, R : Any> Array<out T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {
    for (element in this) {
        val result = transform(element)
        if (result != null) return result
    }
    return null
}
// Returns the first element, or `null` if the array is empty.
*^public fun <T> Array<out T>.firstOrNull(): T? {
    return if (isEmpty()) null else this[0]
}
// Returns the first element, or `null` if the array is empty.
*^public fun ByteArray.firstOrNull(): Byte? {
    return if (isEmpty()) null else this[0]
}
// Returns the first element, or `null` if the array is empty.
*^public fun ShortArray.firstOrNull(): Short? {
    return if (isEmpty()) null else this[0]
}
// Returns the first element, or `null` if the array is empty.
*^public fun IntArray.firstOrNull(): Int? {
    return if (isEmpty()) null else this[0]
}
// Returns the first element, or `null` if the array is empty.
*^public fun LongArray.firstOrNull(): Long? {
    return if (isEmpty()) null else this[0]
}
// Returns the first element, or `null` if the array is empty.
*^public fun FloatArray.firstOrNull(): Float? {
    return if (isEmpty()) null else this[0]
}

```

```

* Returns the first element, or `null` if the array is empty.\n */\npublic fun DoubleArray.firstOrNull(): Double? {\n
return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*/\npublic fun BooleanArray.firstOrNull(): Boolean? {\n  return if (isEmpty()) null else this[0]\n}\n\n/**\n *
Returns the first element, or `null` if the array is empty.\n */\npublic fun CharArray.firstOrNull(): Char? {\n  return
if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if
element was not found.\n */\npublic inline fun <T> Array<out T>.firstOrNull(predicate: (T) -> Boolean): T? {\n
for (element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns the first element
matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun
ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {\n  for (element in this)
if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n */\npublic inline fun ShortArray.firstOrNull(predicate: (Short) ->
Boolean): Short? {\n  for (element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun
IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {\n  for (element in this) if (predicate(element)) return
element\n  return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element
was not found.\n */\npublic inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {\n  for
(element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns the first element
matching the given [predicate], or `null` if element was not found.\n */\npublic
inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {\n  for (element in this) if
(predicate(element)) return element\n  return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n */\npublic inline fun DoubleArray.firstOrNull(predicate: (Double) -
> Boolean): Double? {\n  for (element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun
BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n  for (element in this) if
(predicate(element)) return element\n  return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n */\npublic inline fun CharArray.firstOrNull(predicate: (Char) ->
Boolean): Char? {\n  for (element in this) if (predicate(element)) return element\n  return
null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
[index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.getOrNull(index: Int, defaultValue: (Int) -> T): T? {\n  return if (index >= 0 && index <= lastIndex) get(index)
else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline
fun ByteArray.getOrNull(index: Int, defaultValue: (Int) -> Byte): Byte? {\n  return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.getOrNull(index: Int, defaultValue: (Int) -> Short):
Short? {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of
this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.getOrNull(index: Int, defaultValue: (Int) ->
Int): Int? {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n *
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.getOrNull(index: Int,
defaultValue: (Int) -> Long): Long? {\n  return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.getOrNull(index: Int, defaultValue: (Int) -> Float): Float? {\n  return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n

```

```

*^@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.getOrNull(index: Int, defaultValue: (Int) ->
Double): Double {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n *^@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.getOrNull(index: Int,
defaultValue: (Int) -> Boolean): Boolean {\n    return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling
the [defaultValue] function if the [index] is out of bounds of this array.\n *^@kotlin.internal.InlineOnly\npublic
inline fun CharArray.getOrNull(index: Int, defaultValue: (Int) -> Char): Char {\n    return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if
the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n
*^npublic fun <T> Array<out T>.getOrNull(index: Int): T? {\n    return if (index >= 0 && index <= lastIndex)
get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of
this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n *^npublic fun
ByteArray.getOrNull(index: Int): Byte? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds
of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n *^npublic fun
ShortArray.getOrNull(index: Int): Short? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n
* @sample samples.collections.Collections.Elements.getOrNull\n *^npublic fun IntArray.getOrNull(index: Int):
Int? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n *^npublic fun LongArray.getOrNull(index: Int): Long? {\n
    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*^npublic fun FloatArray.getOrNull(index: Int): Float? {\n    return if (index >= 0 && index <= lastIndex)
get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of
this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n *^npublic fun
DoubleArray.getOrNull(index: Int): Double? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n
* @sample samples.collections.Collections.Elements.getOrNull\n *^npublic fun BooleanArray.getOrNull(index:
Int): Boolean? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n *^npublic fun CharArray.getOrNull(index: Int): Char?
{\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n *^npublic fun <@kotlin.internal.OnlyInputTypes T>
Array<out T>.indexOf(element: T): Int {\n    if (element == null) {\n        for (index in indices) {\n            if
(this[index] == null) {\n                return index\n            }\n        }\n    } else {\n        for (index in indices) {\n
            if (element == this[index]) {\n                return index\n            }\n        }\n    }\n    return -1\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n *^npublic fun ByteArray.indexOf(element:
Byte): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n
return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n *^npublic fun
ShortArray.indexOf(element:
Short): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n
return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n *^npublic fun
IntArray.indexOf(element: Int): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return
index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain
element.\n *^npublic fun LongArray.indexOf(element: Long): Int {\n    for (index in indices) {\n        if (element ==

```

this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element:

T') to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))\n\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6")\n\npublic fun

FloatArray.indexOf(element: Float): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T') to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))\n\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6")\n\npublic fun

DoubleArray.indexOf(element: Double): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n\npublic fun BooleanArray.indexOf(element: Boolean): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n\npublic fun CharArray.indexOf(element: Char): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun <T> Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun

ByteArray.indexOfFirst(predicate: (Byte) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun ShortArray.indexOfFirst(predicate: (Short) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun

LongArray.indexOfFirst(predicate: (Long) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun FloatArray.indexOfFirst(predicate: (Float) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun DoubleArray.indexOfFirst(predicate: (Double) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun BooleanArray.indexOfFirst(predicate: (Boolean) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun <T> Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int {\n for (index in

```

indices.reversed()) {\n    if (predicate(this[index]))
{\n        return index\n    }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the
given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int {\n    for (index
in indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -
1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain
such element.\n */\npublic inline fun LongArray.indexOfLast(predicate: (Long) -> Boolean): Int {\n    for (index in
indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such
element.\n */\npublic inline fun FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {\n    for (index in
indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such
element.\n */\npublic inline fun DoubleArray.indexOfLast(predicate:
(Double) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if (predicate(this[index])) {\n            return
index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -
1 if the array does not contain such element.\n */\npublic inline fun BooleanArray.indexOfLast(predicate: (Boolean)
-> Boolean): Int {\n    for (index in indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n
        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n */\npublic inline fun CharArray.indexOfLast(predicate: (Char) -> Boolean):
Int {\n    for (index in indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n        }\n
    }\n    return -1\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the array is empty.\n
 */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.last(): T {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.last(): Byte {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n * @throws NoSuchElementException if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.last(): Short {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n * @throws NoSuchElementException if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun IntArray.last(): Int {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the
array is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun LongArray.last():
Long {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the array is
empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun FloatArray.last(): Float {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.last(): Double
{\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the array is

```

```

empty.\n * \n * @sample samples.collections.Collections.Elements.last\n *^\npublic fun BooleanArray.last():
Boolean {\n if (isEmpty())\n     throw NoSuchElementException("Array is empty.")\n return
this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n * @sample samples.collections.Collections.Elements.last\n *^\npublic fun CharArray.last(): Char {\n
if (isEmpty())\n     throw NoSuchElementException("Array is empty.")\n return this[lastIndex]\n}\n\n/**\n *
Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such
element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n *^\npublic inline fun <T>
Array<out T>.last(predicate:
(T) -> Boolean): T {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if
(predicate(element)) return element\n     }\n throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ByteArray.last(predicate: (Byte) -> Boolean):
Byte {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if (predicate(element)) return
element\n     }\n throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ShortArray.last(predicate:
(Short) -> Boolean): Short {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if
(predicate(element)) return element\n     }\n throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun IntArray.last(predicate: (Int) -> Boolean): Int
{\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if (predicate(element)) return
element\n     }\n throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun LongArray.last(predicate:
(Long) -> Boolean): Long {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if
(predicate(element)) return element\n     }\n throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun FloatArray.last(predicate: (Float) -> Boolean):
Float {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if (predicate(element))
return element\n     }\n throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline
fun DoubleArray.last(predicate: (Double) -> Boolean): Double {\n for (index in this.indices.reversed()) {\n
val element = this[index]\n     if (predicate(element)) return element\n     }\n throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n *
\n * @sample samples.collections.Collections.Elements.last\n *^\npublic inline fun BooleanArray.last(predicate:
(Boolean) -> Boolean): Boolean {\n for (index in this.indices.reversed()) {\n     val element = this[index]\n     if
(predicate(element)) return element\n     }\n throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n

```

```

*/\npublic inline fun CharArray.last(predicate: (Char) -> Boolean): Char {\n  for (index in this.indices.reversed())
{\n    val element = this[index]\n    if (predicate(element)) return element\n  }\n  throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns last index
of [element], or -1 if the array does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T>
Array<out T>.lastIndexOf(element: T): Int {\n  if (element == null) {\n    for (index in indices.reversed()) {\n
  if (this[index] == null) {\n      return index\n    }\n  }\n } else {\n  for (index in
indices.reversed()) {\n    if (element == this[index]) {\n      return index\n    }\n  }\n }\n }\n
return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
ByteArray.lastIndexOf(element: Byte): Int
{\n  for (index in indices.reversed()) {\n    if (element == this[index]) {\n      return index\n    }\n  }\n
return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
ShortArray.lastIndexOf(element: Short): Int {\n  for (index in indices.reversed()) {\n    if (element ==
this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns last index of [element], or -1 if
the array does not contain element.\n */\npublic fun IntArray.lastIndexOf(element: Int): Int {\n  for (index in
indices.reversed()) {\n    if (element == this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n
* Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
LongArray.lastIndexOf(element: Long): Int {\n  for (index in indices.reversed()) {\n    if (element ==
this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n
*/\n\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed
soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element:
T)' to get the same search behavior as in a list.", ReplaceWith("indexOfLast { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6")\npublic fun
FloatArray.lastIndexOf(element: Float): Int {\n  for (index in indices.reversed()) {\n    if (element ==
this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns last index of [element], or -1 if
the array does not contain element.\n */\n\n@Deprecated("The function has unclear behavior when searching for NaN
or zero values and will be removed soon. Use 'indexOfLast { it == element }' instead to continue using this
behavior, or '.asList().lastIndexOf(element:
T)' to get the same search behavior as in a list.", ReplaceWith("indexOfLast { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6")\npublic fun
DoubleArray.lastIndexOf(element: Double): Int {\n  for (index in indices.reversed()) {\n    if (element ==
this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns last index of [element], or -1 if
the array does not contain element.\n */\npublic fun BooleanArray.lastIndexOf(element: Boolean): Int {\n  for
(index in indices.reversed()) {\n    if (element == this[index]) {\n      return index\n    }\n  }\n  return -
1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
CharArray.lastIndexOf(element: Char): Int {\n  for (index in indices.reversed()) {\n    if (element == this[index])
{\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n\n@sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.lastOrNull(): T? {\n  return if
(isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n\n@sample
samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.lastOrNull(): Byte? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n\n@sample
samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.lastOrNull(): Short? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n\n@sample
samples.collections.Collections.Elements.last\n */\npublic fun IntArray.lastOrNull(): Int? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the
array is empty.\n */\n\n@sample samples.collections.Collections.Elements.last\n */\npublic fun
LongArray.lastOrNull(): Long? {\n  return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last

```



```

element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic
fun FloatArray.lastOrNull(): Float? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last
element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic
fun DoubleArray.lastOrNull(): Double? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the
last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\npublic fun BooleanArray.lastOrNull(): Boolean? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last element, or `null` if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic fun CharArray.lastOrNull(): Char? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Array<out T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun ByteArray.lastOrNull(predicate: (Byte) ->
Boolean): Byte? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun ShortArray.lastOrNull(predicate: (Short) ->
Boolean): Short? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun IntArray.lastOrNull(predicate: (Int) ->
Boolean): Int? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun LongArray.lastOrNull(predicate:
(Long) -> Boolean): Long? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.lastOrNull(predicate: (Float) ->
Boolean): Float? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun DoubleArray.lastOrNull(predicate: (Double) ->
Boolean): Double? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\npublic inline fun BooleanArray.lastOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n    for (index in this.indices.reversed()) {\n
        val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun CharArray.lastOrNull(predicate: (Char) ->
Boolean): Char? {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns a random element from this array.\n
*/\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n * \n * @since Kotlin("1.3")\n */\n @kotlin.internal.InlineOnly\n public

```

```

inline fun <T> Array<out T>.random(): T {\n    return random(Random)\n}\n\n/**\n * Returns a random element
from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.random(): Byte {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.random(): Short {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.random(): Int {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.random(): Long {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.random(): Float {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.random(): Double {\n    return random(Random)\n}\n\n/**\n * Returns a random element
from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.random(): Boolean {\n
    return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.random(): Char {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is
empty.\n */\n\n@SinceKotlin("1.3")\npublic fun <T> Array<out T>.random(random: Random): T {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic
fun ByteArray.random(random: Random): Byte {\n    if (isEmpty())\n        throw NoSuchElementException("Array
is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the
specified source of randomness.\n * \n * @throws NoSuchElementException if this array is
empty.\n */\n\n@SinceKotlin("1.3")\npublic fun ShortArray.random(random: Random): Short {\n    if (isEmpty())\n
        throw NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns
a random element from this array using the specified source of randomness.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic fun IntArray.random(random:
Random): Int {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic
fun LongArray.random(random: Random): Long {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source
of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n\n@SinceKotlin("1.3")\npublic fun FloatArray.random(random: Random): Float {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic fun DoubleArray.random(random: Random): Double {\n
    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic
fun BooleanArray.random(random: Random): Boolean {\n    if (isEmpty())\n        throw

```

```

NoSuchElementException("Array is empty.\n")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws\n NoSuchElementException if this array is empty.\n */\n\n@SinceKotlin("1.3")\npublic fun\n CharArray.random(random: Random): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Array is\n empty.\n")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if\n this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.randomOrNull(): Byte? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random\n element from this array, or `null`\n if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.randomOrNull(): Short? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.randomOrNull(): Int? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random\n element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.randomOrNull(): Long? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.randomOrNull():\n Float? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this\n array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.randomOrNull(): Double? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.randomOrNull(): Boolean? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a\n random element from this array using the specified source of\n randomness, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out\n T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return\n get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of\n randomness, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun\n ByteArray.randomOrNull(random: Random): Byte? {\n    if (isEmpty())\n        return null\n    return\n get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of\n randomness, or `null` if this array is empty.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun\n ShortArray.randomOrNull(random: Random): Short? {\n    if (isEmpty())\n        return null\n    return\n get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from

```

```

this array using the specified source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntArray.randomOrNull(random: Random): Int? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongArray.randomOrNull(random: Random): Long? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
FloatArray.randomOrNull(random: Random): Float? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is
empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
DoubleArray.randomOrNull(random: Random): Double? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
BooleanArray.randomOrNull(random: Random): Boolean? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharArray.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return
null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the
array is empty or has more than one element.\n
*\npublic fun <T> Array<out T>.single(): T {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\npublic fun ByteArray.single(): Byte {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n
else -> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the
single element, or throws an exception if the array is empty or has more than one element.\n
*\npublic fun ShortArray.single(): Short {\n return when (size) {\n 0 -> throw NoSuchElementException("Array
is empty.")\n 1 -> this[0]\n else -> throw IllegalArgumentException("Array has more than one
element.")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more
than one element.\n
*\npublic fun IntArray.single(): Int {\n return when (size) {\n 0 -> throw
NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\npublic fun LongArray.single(): Long {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n
else -> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the
single element, or throws an exception if the array is empty or has more than one element.\n
*\npublic
fun FloatArray.single(): Float {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is
empty.")\n 1 -> this[0]\n else -> throw IllegalArgumentException("Array has more than one element.")\n
}\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one
element.\n
*\npublic fun DoubleArray.single(): Double {\n return when (size) {\n 0 -> throw
NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\npublic fun BooleanArray.single():

```

```

Boolean {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun CharArray.single(): Char {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as T
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Byte
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Short
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Int
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Long
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Float
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw

```



```

if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or
`null` if element was not found or more than one element was found.\n */\npublic inline fun
LongArray.singleOrNull(predicate: (Long) -> Boolean): Long? {\n    var single: Long? = null\n    var found =
false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single =
element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n */\npublic inline fun FloatArray.singleOrNull(predicate:
(Float) -> Boolean): Float? {\n    var single: Float? = null\n    var found = false\n    for (element in this) {\n        if
(predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n
if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or
`null` if element was not found or more than one element was found.\n */\npublic inline fun DoubleArray.singleOrNull(predicate: (Double) -> Boolean): Double? {\n    var single: Double? = null\n    var found
= false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single =
element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n */\npublic inline fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n    var single: Boolean? =
null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n
            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n *
Returns the single element matching the given [predicate], or `null` if element was not found or more than one
element was found.\n */\npublic inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {\n    var
single: Char? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if
(found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return
single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n */\n * @throws
IllegalArgumentException if
[n] is negative.\n */\n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T>
Array<out T>.drop(n: Int): List<T> {\n    require(n >= 0) { "\"Requested element count $n is less than zero.\" }\n
return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n]
elements.\n */\n * @throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.drop(n: Int): List<Byte> {\n
require(n >= 0) { "\"Requested element count $n is less than zero.\" }\n    return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun ShortArray.drop(n: Int): List<Short> {\n
require(n >= 0) { "\"Requested element count $n is less than zero.\" }\n
}\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n]
elements.\n */\n * @throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun IntArray.drop(n: Int): List<Int> {\n
require(n >= 0) { "\"Requested element count $n is less than zero.\" }\n    return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun LongArray.drop(n: Int): List<Long> {\n
require(n >= 0) { "\"Requested element count $n is less than zero.\" }\n    return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic fun FloatArray.drop(n: Int): List<Float> {\n    require(n >= 0) { "\"Requested element count $n is less
than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements

```

except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun DoubleArray.drop(n: Int): List<Double> {\n
require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun BooleanArray.drop(n: Int): List<Boolean>
{\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun CharArray.drop(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n
return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n]
elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun <T> Array<out T>.dropLast(n: Int): List<T>
{\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun ByteArray.dropLast(n: Int): List<Byte> {\n require(n >= 0) { \"Requested element count \$n is less
than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun ShortArray.dropLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun IntArray.dropLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list
containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
\n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun LongArray.dropLast(n: Int):
List<Long> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun FloatArray.dropLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun DoubleArray.dropLast(n:
Int): List<Double> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun BooleanArray.dropLast(n: Int):
List<Boolean> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun CharArray.dropLast(n: Int): List<Char> {\n
require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last

elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun <T> Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun ByteArray.dropLastWhile(predicate: (Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun ShortArray.dropLastWhile(predicate: (Short) -> Boolean): List<Short> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun IntArray.dropLastWhile(predicate: (Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun LongArray.dropLastWhile(predicate: (Long) -> Boolean): List<Long> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun FloatArray.dropLastWhile(predicate: (Float) -> Boolean): List<Float> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun DoubleArray.dropLastWhile(predicate: (Double) -> Boolean): List<Double> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun <T> Array<out T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n var yielding = false\n val list = ArrayList<T>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample

samples.collections.Collections.Transformations.drop\n *\npublic inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {\n var yielding = false\n val list = ArrayList<Byte>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample

```

\n * @sample samples.collections.Collections.Transformations.drop\n *^npublic inline fun
ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {\n  var yielding = false\n  val list =
ArrayList<Short>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))\n      list.add(item)\n    yielding = true\n  }\n  return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *^npublic inline fun IntArray.dropWhile(predicate: (Int) ->
Boolean): List<Int> {\n  var yielding = false\n  val list = ArrayList<Int>()\n  for (item in this)\n    if
(yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding = true\n
    }\n  return list\n}\n\n/**\n * Returns a list containing all elements except
first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *^npublic inline fun LongArray.dropWhile(predicate:
(Long) -> Boolean): List<Long> {\n  var yielding = false\n  val list = ArrayList<Long>()\n  for (item in this)\n
if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding =
true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *^npublic inline fun
FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n  var yielding = false\n  val list =
ArrayList<Float>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))\n
      list.add(item)\n    yielding = true\n  }\n  return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *^npublic inline fun DoubleArray.dropWhile(predicate:
(Double) -> Boolean): List<Double> {\n  var yielding = false\n  val list = ArrayList<Double>()\n  for (item in
this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding =
true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *^npublic
inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n  var yielding = false\n
val list = ArrayList<Boolean>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if
(!predicate(item)) {\n      list.add(item)\n      yielding = true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all elements
except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *^npublic inline fun CharArray.dropWhile(predicate:
(Char) -> Boolean): List<Char> {\n  var yielding = false\n  val list = ArrayList<Char>()\n  for (item in this)\n
if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding =
true\n    }\n  return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n *^npublic inline fun <T> Array<out
T>.filter(predicate: (T) -> Boolean): List<T> {\n  return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns
a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *^npublic inline fun ByteArray.filter(predicate: (Byte) -> Boolean): List<Byte> {\n  return
filterTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *^npublic inline fun
ShortArray.filter(predicate: (Short) -> Boolean): List<Short> {\n  return filterTo(ArrayList<Short>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *^npublic inline fun IntArray.filter(predicate: (Int) -> Boolean):
List<Int> {\n  return filterTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *^npublic inline
fun LongArray.filter(predicate: (Long) -> Boolean): List<Long> {\n  return filterTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *^npublic inline fun FloatArray.filter(predicate: (Float) ->

```

```

Boolean): List<Float> {\n  return filterTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\npublic inline fun DoubleArray.filter(predicate: (Double) -> Boolean): List<Double> {\n  return
filterTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun
BooleanArray.filter(predicate: (Boolean) -> Boolean): List<Boolean> {\n  return filterTo(ArrayList<Boolean>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n
* \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun CharArray.filter(predicate:
(Char) -> Boolean): List<Char> {\n  return filterTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list
containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an
element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun <T> Array<out
T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n  return filterIndexedTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
*\npublic inline fun ByteArray.filterIndexed(predicate: (index: Int, Byte) -> Boolean): List<Byte> {\n  return
filterIndexedTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun ShortArray.filterIndexed(predicate:
(index: Int, Short) -> Boolean): List<Short> {\n  return filterIndexedTo(ArrayList<Short>(), predicate)\n}\n\n/**\n
* Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun
IntArray.filterIndexed(predicate:
(index: Int, Int) -> Boolean): List<Int> {\n  return filterIndexedTo(ArrayList<Int>(), predicate)\n}\n\n/**\n *
Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun
LongArray.filterIndexed(predicate: (index: Int, Long) -> Boolean): List<Long> {\n  return
filterIndexedTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun FloatArray.filterIndexed(predicate:
(index: Int, Float) ->
Boolean): List<Float> {\n  return filterIndexedTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list
containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an
element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun DoubleArray.filterIndexed(predicate:
(index: Int, Double) -> Boolean): List<Double> {\n  return filterIndexedTo(ArrayList<Double>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic
inline fun BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {\n
  return filterIndexedTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element
itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

```

```

samples.collections.Collections.Filtering.filterIndexed\n * \npublic inline fun CharArray.filterIndexed(predicate:
(index: Int, Char) -> Boolean): List<Char> {\n  return filterIndexedTo(ArrayList<Char>(), predicate)\n}\n\n/**\n *
Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that
takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the
element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <T, C :
MutableCollection<in T>> Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean):
C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n}
return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n *
\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate:
(index: Int, Byte) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <C :
MutableCollection<in Short>> ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) ->
Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <C : MutableCollection<in Int>>
IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) -> Boolean): C {\n  forEachIndexed { index,
element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n *
Appends all elements
matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an
element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {\n  forEachIndexed {
index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n *
\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination:
C, predicate: (index: Int, Float) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index,
element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the
given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and
the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <C : MutableCollection<in
Double>> DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {\n  forEachIndexed
{ index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns
the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <C : MutableCollection<in
Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {\n  forEachIndexed
{ index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *

```

```

@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*/\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterIndexedTo(destination: C, predicate:
(index: Int, Char) -> Boolean): C {\n    forEachIndexed { index, element ->\n        if (predicate(index, element))
destination.add(element)\n
}\n    return destination\n}\n\n/**\n * Returns a list containing all elements that are instances of specified type
parameter R.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun
<reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n    return
filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C :
MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {\n    for (element in this) if (element is
R) destination.add(element)\n    return destination\n}\n\n/**\n * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T>
Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {\n
    return filterNotTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun
ByteArray.filterNot(predicate: (Byte) -> Boolean): List<Byte> {\n    return filterNotTo(ArrayList<Byte>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun ShortArray.filterNot(predicate: (Short) ->
Boolean): List<Short> {\n    return filterNotTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun IntArray.filterNot(predicate: (Int) -> Boolean):
List<Int> {\n    return filterNotTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing all elements
not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun
LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {\n    return filterNotTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filterNot(predicate: (Float) ->
Boolean): List<Float> {\n    return filterNotTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun DoubleArray.filterNot(predicate: (Double) ->
Boolean): List<Double> {\n    return filterNotTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic
inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {\n    return
filterNotTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun
CharArray.filterNot(predicate: (Char) -> Boolean): List<Char> {\n    return filterNotTo(ArrayList<Char>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements that are not `null`.\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNull\n */\npublic fun <T : Any> Array<out T?>.filterNotNull():
List<T> {\n    return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the
given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C
: MutableCollection<in T>, T : Any> Array<out T?>.filterNotNullTo(destination: C): C {\n    for (element in this) if
(element != null) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterNotTo(destination: C, predicate: (T) ->
Boolean): C {\n    for (element in this) if (!predicate(element)) destination.add(element)\n    return

```

```

destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in
Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -
> Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Int>>
IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {\n  for (element in this) if
(!predicate(element))
destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not matching the given
[predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) ->
Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in
Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterNotTo(destination: C, predicate:
(Boolean) -> Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in
Char>> CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterTo(destination: C, predicate: (T) ->
Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n
return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n
\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in
Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {\n  for (element in this) if
(predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) ->
Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic
inline fun <C : MutableCollection<in Int>> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {\n
for (element in this) if (predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends
all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Long>>

```

```

LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if
    (predicate(element)) destination.add(element)
    return destination
}

* Appends all elements matching
the given [predicate] to the given [destination].

@sample samples.collections.Collections.Filtering.filterTo

public inline fun <C : MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) ->
Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return
    destination
}

* Appends all elements matching the given [predicate] to the given [destination].

@sample
samples.collections.Collections.Filtering.filterTo

public inline fun <C : MutableCollection<in Double>>
DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if
    (predicate(element)) destination.add(element)
    return destination
}

* Appends all elements matching
the given [predicate] to the given [destination].

@sample samples.collections.Collections.Filtering.filterTo

public inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterTo(destination: C, predicate:
(Boolean) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return
    destination
}

* Appends all elements matching the given [predicate] to the given [destination].

@sample samples.collections.Collections.Filtering.filterTo

public inline fun <C : MutableCollection<in
Char>> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if
    (predicate(element)) destination.add(element)
    return destination
}

Returns a list containing elements
at indices in the specified [indices] range.

public fun <T> Array<out T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive +
    1).asList()
}

Returns a list containing elements at indices in the specified [indices] range.

public fun ByteArray.slice(indices: IntRange): List<Byte> {
    if (indices.isEmpty()) return listOf()
    return
    copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

Returns a list containing elements at
indices in the specified [indices] range.

public fun ShortArray.slice(indices: IntRange): List<Short> {
    if
    (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive
    + 1).asList()
}

Returns a list containing elements at indices in the specified [indices] range.

public fun IntArray.slice(indices: IntRange): List<Int> {
    if (indices.isEmpty()) return listOf()
    return
    copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

Returns a list containing elements at
indices in the specified [indices] range.

public fun LongArray.slice(indices: IntRange): List<Long> {
    if
    (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive +
    1).asList()
}

Returns a list containing elements at
indices in the specified [indices] range.

public fun FloatArray.slice(indices: IntRange): List<Float> {
    if (indices.isEmpty()) return listOf()
    return
    copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

Returns a list containing elements at
indices in the specified [indices] range.

public fun DoubleArray.slice(indices: IntRange):
List<Double> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive
    + 1).asList()
}

Returns a list containing elements at indices in the specified [indices] range.

public fun BooleanArray.slice(indices: IntRange): List<Boolean> {
    if (indices.isEmpty()) return listOf()
    return
    copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

Returns a list containing elements at
indices in the specified [indices] range.

public fun CharArray.slice(indices: IntRange): List<Char> {
    if
    (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive +
    1).asList()
}

Returns a list containing elements at specified [indices].

public fun <T> Array<out
T>.slice(indices: Iterable<Int>): List<T> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0)
    return emptyList()
    val list = ArrayList<T>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Returns a list containing elements at specified [indices].

public fun ByteArray.slice(indices: Iterable<Int>): List<Byte> {
    val size =
    indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Byte>(size)
    for
    (index in indices) {
        list.add(get(index))
    }
    return list
}

Returns a list containing elements at
specified [indices].

public fun ShortArray.slice(indices: Iterable<Int>): List<Short> {
    val size =
    indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Short>(size)
}

```

```

for (index in indices) {\n    list.add(get(index))\n } return list\n\n/**\n * Returns a list containing
elements at specified [indices].\n */\npublic fun IntArray.slice(indices: Iterable<Int>): List<Int> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n
    val list = ArrayList<Int>(size)\n    for (index in indices) {\n        list.add(get(index))\n    } return
list\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun LongArray.slice(indices:
Iterable<Int>): List<Long> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return
emptyList()\n    val list = ArrayList<Long>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }
return list\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun
FloatArray.slice(indices: Iterable<Int>): List<Float> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size
== 0) return emptyList()\n    val list = ArrayList<Float>(size)\n    for (index in indices) {\n
list.add(get(index))\n } return list\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun DoubleArray.slice(indices: Iterable<Int>): List<Double> {\n    val size
= indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Double>(size)\n
for (index in indices) {\n        list.add(get(index))\n    } return list\n\n/**\n * Returns a list containing
elements at specified [indices].\n */\npublic fun BooleanArray.slice(indices: Iterable<Int>): List<Boolean> {\n    val
size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =
ArrayList<Boolean>(size)\n    for (index in indices) {\n        list.add(get(index))\n    } return list\n\n/**\n *
Returns a list containing elements at specified [indices].\n */\npublic fun CharArray.slice(indices: Iterable<Int>):
List<Char> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =
ArrayList<Char>(size)\n    for (index in indices) {\n        list.add(get(index))\n    } return list\n\n/**\n *
Returns an array containing elements of this array at specified
[indices].\n */\npublic fun <T> Array<T>.sliceArray(indices: Collection<Int>): Array<T> {\n    val result =
arrayOfNulls(this, indices.size)\n    var targetIndex = 0\n    for (sourceIndex in indices) {\n
result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns an array containing elements
of this array at specified [indices].\n */\npublic fun ByteArray.sliceArray(indices: Collection<Int>): ByteArray {\n
val result = ByteArray(indices.size)\n    var targetIndex = 0\n    for (sourceIndex in indices) {\n
result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns an array containing elements
of this array at specified [indices].\n */\npublic fun ShortArray.sliceArray(indices: Collection<Int>): ShortArray {\n
val result = ShortArray(indices.size)\n    var targetIndex = 0\n    for (sourceIndex in indices) {\n
result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns
an array containing elements of this array at specified [indices].\n */\npublic fun IntArray.sliceArray(indices:
Collection<Int>): IntArray {\n    val result = IntArray(indices.size)\n    var targetIndex = 0\n    for (sourceIndex in
indices) {\n        result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns an array
containing elements of this array at specified [indices].\n */\npublic fun LongArray.sliceArray(indices:
Collection<Int>): LongArray {\n    val result = LongArray(indices.size)\n    var targetIndex = 0\n    for (sourceIndex
in indices) {\n        result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns an array
containing elements of this array at specified [indices].\n */\npublic fun FloatArray.sliceArray(indices:
Collection<Int>): FloatArray {\n    val result = FloatArray(indices.size)\n    var targetIndex = 0\n    for (sourceIndex
in indices) {\n        result[targetIndex++] = this[sourceIndex]\n } return result\n\n/**\n * Returns an array
containing elements of this array at specified [indices].\n */\npublic fun DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray {\n    val result = DoubleArray(indices.size)\n
var targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n } return
result\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
BooleanArray.sliceArray(indices: Collection<Int>): BooleanArray {\n    val result = BooleanArray(indices.size)\n
var targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n } return
result\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
CharArray.sliceArray(indices: Collection<Int>): CharArray {\n    val result = CharArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n

```



```

    result[targetIndex++] = this[sourceIndex]\n    }\n    return result\n}\n\n/**\n * Returns an array containing
elements at indices in the specified [indices] range.\n */\npublic fun <T> Array<T>.sliceArray(indices: IntRange):
Array<T> {\n    if (indices.isEmpty()) return copyOfRange(0, 0)\n    return copyOfRange(indices.start,
indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices]
range.\n */\npublic fun ByteArray.sliceArray(indices: IntRange): ByteArray {\n    if (indices.isEmpty()) return
ByteArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array
containing elements at indices in the specified [indices] range.\n */\npublic fun ShortArray.sliceArray(indices:
IntRange): ShortArray {\n    if (indices.isEmpty()) return ShortArray(0)\n    return copyOfRange(indices.start,
indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices]
range.\n */\npublic fun IntArray.sliceArray(indices: IntRange): IntArray {\n    if (indices.isEmpty()) return
IntArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array
containing elements at indices in the specified [indices] range.\n */\npublic fun LongArray.sliceArray(indices:
IntRange): LongArray {\n    if (indices.isEmpty()) return LongArray(0)\n    return copyOfRange(indices.start,
indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices]
range.\n */\npublic fun FloatArray.sliceArray(indices: IntRange): FloatArray {\n    if (indices.isEmpty()) return
FloatArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array
containing elements at indices in the specified [indices] range.\n */\npublic fun DoubleArray.sliceArray(indices:
IntRange): DoubleArray {\n    if (indices.isEmpty()) return DoubleArray(0)\n    return copyOfRange(indices.start,
indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices]
range.\n */\npublic fun BooleanArray.sliceArray(indices: IntRange): BooleanArray {\n    if (indices.isEmpty())
return BooleanArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an
array containing elements at indices in the specified [indices] range.\n */\npublic fun CharArray.sliceArray(indices:
IntRange): CharArray {\n    if (indices.isEmpty()) return CharArray(0)\n    return copyOfRange(indices.start,
indices.endInclusive + 1)\n}\n\n/**\n * Returns a list containing first [n] elements.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.take(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n
    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<T>(n)\n    for (item in this) {\n
list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first
[n] elements.\n */\n * @throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ByteArray.take(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Byte>(n)\n
for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n */\n * @throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.take(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less
than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(this[0])\n    var count = 0\n    val list = ArrayList<Short>(n)\n    for (item in this) {\n        list.add(item)\n
if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n */\n *
@throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.take(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Int>(n)\n
for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample

```

```

samples.collections.Collections.Transformations.take\n *^\npublic fun LongArray.take(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Long>(n)\n for
(item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun FloatArray.take(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n
if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list =
ArrayList<Float>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n
return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic fun
DoubleArray.take(n: Int): List<Double> {\n require(n >= 0) { \"Requested element count $n is less than zero.\"\n
}\n if (n == 0) return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var
count = 0\n val list = ArrayList<Double>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n
break\n }\n return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun BooleanArray.take(n: Int): List<Boolean>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n
>= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Boolean>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n *^\npublic fun CharArray.take(n: Int): List<Char>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n
>= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Char>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns a list containing
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<T>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun ByteArray.takeLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n
if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return
listOf(this[size - 1])\n val list = ArrayList<Byte>(n)\n for (index in size - n until size)\n
list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun ShortArray.takeLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Short>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*^\npublic fun IntArray.takeLast(n: Int): List<Int> {\n require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return
listOf(this[size - 1])\n val list = ArrayList<Int>(n)\n for (index in size - n until size)\n list.add(this[index])\n
return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if

```

```

[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n\npublic fun
LongArray.takeLast(n: Int): List<Long> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\"
}\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(this[size - 1])\n    val list = ArrayList<Long>(n)\n    for (index in size - n until size)\nlist.add(this[index])\n
    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n\npublic fun
FloatArray.takeLast(n: Int): List<Float> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n
    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(this[size - 1])\n    val list = ArrayList<Float>(n)\n    for (index in size - n until size)\nlist.add(this[index])\n
    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic fun DoubleArray.takeLast(n: Int): List<Double>
{\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val
size = size\n
    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Double>(n)\n    for
(index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n]
elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic fun BooleanArray.takeLast(n: Int):
List<Boolean> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return
emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list
= ArrayList<Boolean>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic
fun CharArray.takeLast(n: Int): List<Char> {\n    require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(this[size - 1])\n    val list = ArrayList<Char>(n)\n    for (index in size - n until size)\nlist.add(this[index])\n
    return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n\npublic inline fun <T>
Array<out T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n    for (index in lastIndex downTo 0) {\n        if
(!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic inline fun ByteArray.takeLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
            return drop(index + 1)\n        }\n    }\n    return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n\npublic
inline fun ShortArray.takeLastWhile(predicate: (Short) -> Boolean): List<Short> {\n    for (index in lastIndex
downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic inline fun IntArray.takeLastWhile(predicate: (Int)
-> Boolean): List<Int> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n           
return
drop(index + 1)\n        }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic inline fun LongArray.takeLastWhile(predicate:
(Long) -> Boolean): List<Long> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
            return
drop(index + 1)\n        }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n * \n\npublic
inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {\n    for (index in lastIndex

```

```

downTo 0) {\n    if (!predicate(this[index])) {\n        return drop(index + 1)\n    }\n } return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic
inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {\n    for (index in lastIndex
downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    } return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun
BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    for (index in lastIndex
downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    } return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun CharArray.takeLastWhile(predicate:
(Char) -> Boolean): List<Char> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
return drop(index + 1)\n        }\n    } return toList()\n}\n\n/**\n * Returns a list containing first elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *\npublic
inline fun <T> Array<out T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n    val list = ArrayList<T>()\n    for
(item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    } return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun ByteArray.takeWhile(predicate: (Byte)
-> Boolean): List<Byte> {\n    val list = ArrayList<Byte>()\n    for (item in this) {\n        if (!predicate(item))\n
break\n        list.add(item)\n    } return list\n}\n\n/**\n * Returns a list containing first elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*\npublic inline fun ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {\n    val list =
ArrayList<Short>()\n    for (item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun IntArray.takeWhile(predicate: (Int) ->
Boolean): List<Int> {\n    val list = ArrayList<Int>()\n    for (item in this) {\n        if (!predicate(item))\n
break\n        list.add(item)\n    } return list\n}\n\n/**\n * Returns a list containing first elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *\npublic inline fun
LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {\n    val list = ArrayList<Long>()\n    for (item
in this) {\n        if
(!predicate(item))\n            break\n        list.add(item)\n    } return list\n}\n\n/**\n * Returns a list containing
first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun FloatArray.takeWhile(predicate:
(Float) -> Boolean): List<Float> {\n    val list = ArrayList<Float>()\n    for (item in this) {\n        if
(!predicate(item))\n            break\n        list.add(item)\n    } return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*\npublic inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {\n    val list =
ArrayList<Double>()\n    for (item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n *
\n * @sample samples.collections.Collections.Transformations.take\n *\npublic inline fun
BooleanArray.takeWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    val list = ArrayList<Boolean>()\n
for (item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    } return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun CharArray.takeWhile(predicate: (Char)
-> Boolean): List<Char> {\n    val list = ArrayList<Char>()\n    for (item in this) {\n        if (!predicate(item))\n
break\n        list.add(item)\n    } return list\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic
fun <T> Array<T>.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var
reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n

```

```

    this[index] = this[reverseIndex]\n    this[reverseIndex] = tmp\n    reverseIndex--\n    }\n}\n\n/**\n *
Reverses elements in the array in-place.\n */\npublic fun ByteArray.reverse(): Unit {\n    val midPoint = (size / 2) -
1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp =
this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun ShortArray.reverse(): Unit {\n    val
midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint)
{\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun IntArray.reverse(): Unit
{\n    val midPoint = (size / 2) - 1\n    if (midPoint <
0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses
elements in the array in-place.\n */\npublic fun LongArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if
(midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n *
Reverses elements in the array in-place.\n */\npublic fun FloatArray.reverse(): Unit {\n    val midPoint = (size / 2) -
1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp =
this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array
in-place.\n */\npublic fun DoubleArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0)
return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index]
= this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in
the array in-place.\n */\npublic fun BooleanArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint
< 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses
elements in the array in-place.\n */\npublic fun CharArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if
(midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n    this[reverseIndex] = tmp\n    reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the
specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param
toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun <T> Array<T>.reverse(fromIndex:
Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint =
(fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in
fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n
this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements
of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to
reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic
fun ByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var
reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses
elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to
reverse.\n * \n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws

```

```

IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun
ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n  val midPoint = (fromIndex + toIndex) / 2\n  if (fromIndex == midPoint) return\n  var reverseIndex =
toIndex - 1\n  for (index in fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] =
this[reverseIndex]\n    this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun
IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n  val midPoint = (fromIndex + toIndex) / 2\n  if (fromIndex == midPoint) return\n  var reverseIndex =
toIndex - 1\n  for (index in fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] =
this[reverseIndex]\n    this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException
if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun LongArray.reverse(fromIndex:
Int, toIndex: Int): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val midPoint =
(fromIndex + toIndex) / 2\n  if (fromIndex == midPoint) return\n  var reverseIndex = toIndex - 1\n  for (index in
fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] = this[reverseIndex]\n
this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun FloatArray.reverse(fromIndex:
Int, toIndex: Int): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val midPoint =
(fromIndex + toIndex) / 2\n  if (fromIndex == midPoint) return\n  var reverseIndex = toIndex - 1\n  for (index in
fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] = this[reverseIndex]\n
this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val midPoint
= (fromIndex + toIndex) / 2\n  if (fromIndex == midPoint) return\n  var reverseIndex = toIndex - 1\n  for (index
in fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] = this[reverseIndex]\n
this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.4")\npublic fun BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit
{\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val midPoint = (fromIndex + toIndex) / 2\n
  if (fromIndex == midPoint) return\n  var reverseIndex = toIndex - 1\n
  for (index in fromIndex until midPoint) {\n    val tmp = this[index]\n    this[index] = this[reverseIndex]\n
this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *

```

```

or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *^@SinceKotlin("1.4")\npublic fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if
(fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n
        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Returns a list with elements in reversed
order.\n *^npublic fun <T> Array<out T>.reversed(): List<T> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun ByteArray.reversed(): List<Byte> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun ShortArray.reversed(): List<Short> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun IntArray.reversed(): List<Int> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements
in reversed order.\n *^npublic fun LongArray.reversed(): List<Long> {\n    if (isEmpty()) return emptyList()\n
val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed
order.\n *^npublic fun FloatArray.reversed(): List<Float> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun DoubleArray.reversed(): List<Double> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun BooleanArray.reversed(): List<Boolean> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*^npublic fun CharArray.reversed(): List<Char> {\n    if (isEmpty()) return
emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns an array with
elements of this array in reversed order.\n *^npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if
(isEmpty()) return this\n    val result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in
0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this
array in reversed order.\n *^npublic fun ByteArray.reversedArray(): ByteArray {\n    if (isEmpty()) return this\n
val result = ByteArray(size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] =
this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n *^npublic fun
ShortArray.reversedArray(): ShortArray {\n    if (isEmpty()) return this\n    val result = ShortArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in
reversed order.\n *^npublic fun IntArray.reversedArray(): IntArray {\n    if (isEmpty()) return this\n    val result =
IntArray(size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return
result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n *^npublic fun
LongArray.reversedArray(): LongArray {\n    if (isEmpty()) return this\n    val result = LongArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n *^npublic fun FloatArray.reversedArray():
FloatArray {\n    if (isEmpty()) return this\n    val result = FloatArray(size)\n    val lastIndex = lastIndex\n    for (i
in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n *^npublic fun
DoubleArray.reversedArray(): DoubleArray {\n    if (isEmpty()) return this\n    val result = DoubleArray(size)\n
val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n *^npublic fun BooleanArray.reversedArray():
BooleanArray {\n    if (isEmpty()) return this\n    val result = BooleanArray(size)\n    val lastIndex = lastIndex\n
for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with
elements of this array in reversed order.\n *^npublic fun CharArray.reversedArray(): CharArray {\n    if (isEmpty())

```



```
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.shuffle(random: Random): Unit {\n    for (i in lastIndex\n        downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] = this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the\n * source of randomness.\n * \n * See:
```

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n

```
*\n@SinceKotlin("1.4")\npublic fun BooleanArray.shuffle(random: Random): Unit {\n    for (i in\n        lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] = this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]\n * instance as the source of randomness.\n * \n * See:
```

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n

```
*\n@SinceKotlin("1.4")\npublic fun CharArray.shuffle(random: Random): Unit {\n    for (i in lastIndex downTo\n        1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] = this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Sorts elements in the array in-place according to natural sort order of the value returned by specified\n * [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each\n * other after sorting.\n * \npublic inline fun <T, R : Comparable<R>> Array<out T>.sortBy(crossinline selector: (T) -\n    > R?): Unit {\n    if (size >
```

```
1) sortWith(compareBy(selector))\n}\n\n/**\n * Sorts elements in the array in-place descending according to\n * natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that\n * equal elements preserve their order relative to each other after sorting.\n * \npublic inline fun <T, R :
```

```
Comparable<R>> Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit {\n    if (size > 1)\n        sortWith(compareByDescending(selector))\n}\n\n/**\n * Sorts elements in the array in-place descending according\n * to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to\n * each other after sorting.\n * \npublic fun <T : Comparable<T>> Array<out T>.sortDescending(): Unit {\n    sortWith(reverseOrder())\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural
```

```
sort order.\n * \npublic fun ByteArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort
```

```
order.\n * \npublic fun ShortArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n * \npublic fun
```

```
IntArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements\n * in the array in-place descending according to their natural sort order.\n * \npublic fun LongArray.sortDescending():\n    Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements in the array in-place
```

```
descending according to their natural sort order.\n * \npublic fun FloatArray.sortDescending(): Unit {\n    if (size >\n        1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements in the array in-place descending according to\n * their natural sort order.\n * \npublic fun
```

```
DoubleArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts\n * elements in the array in-place descending according to their natural sort order.\n * \npublic fun
```

```
CharArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Returns a list\n * of all elements sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements\n * preserve their order relative to each other after sorting.\n * \npublic fun <T : Comparable<T>> Array<out
```

```
T>.sorted(): List<T> {\n    return sortedArray().asList()\n}\n\n/**\n * Returns a list of all elements sorted according\n * to their natural sort order.\n * \npublic fun ByteArray.sorted(): List<Byte> {\n    return toTypedArray().apply {\n        sort()\n    }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \npublic
```

```
fun ShortArray.sorted(): List<Short> {\n    return toTypedArray().apply\n        {\n            sort()\n        }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \n
```

```
public fun IntArray.sorted(): List<Int> {\n    return toTypedArray().apply {\n        sort()\n    }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \npublic fun LongArray.sorted():
```

```
List<Long> {\n    return toTypedArray().apply {\n        sort()\n    }.asList()\n}\n\n/**\n * Returns a list of all elements sorted\n * according to their natural sort order.\n * \npublic fun FloatArray.sorted(): List<Float> {\n    return
```


of all elements sorted according to natural sort order of the value returned by specified [selector] function.

The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <T, R : Comparable<R>> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedBy`

```
public inline fun <R : Comparable<R>> CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {
    return sortedWith(compareBy(selector))
}
```

Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.

The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

`@sample samples.collections.Collections.Sorting.sortedByDescending`

```
public inline fun <T, R : Comparable<R>> Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareByDescending(selector))
}
```

Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedByDescending`

```
public inline fun <R : Comparable<R>> ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> {
    return sortedWith(compareByDescending(selector))
}
```

Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.

`@sample samples.collections.Collections.Sorting.sortedByDescending`

```
public inline fun <R : Comparable<R>> ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> {
    return sortedWith(compareByDescending(selector))
}
```

Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.

Comparable<R>> IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedDescending(): List<Byte> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedDescending(): List<Short> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedDescending(): List<Int> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedDescending(): List<Long> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedDescending(): List<Float> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedDescending(): List<Double> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedDescending(): List<Char> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedWith(comparator: Comparator<in T>): List<T> {\n return sortedArrayWith(comparator).asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {\n return

`toTypedArray().apply { sortBy(comparator) }.asList()` Returns a list of all elements sorted according to the specified [comparator].

`FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float>` Returns a list of all elements sorted according to the specified [comparator].

`DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double>` Returns a list of all elements sorted according to the specified [comparator].

`BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean>` Returns a list of all elements sorted according to the specified [comparator].

`CharArray.sortedWith(comparator: Comparator<in Char>): List<Char>` Returns a [List] that wraps the original array.

`<T> Array<out T>.asList(): List<T>` Returns a [List] that wraps the original array.

`ByteArray.asList(): List<Byte>` Returns a [List] that wraps the original array.

`ShortArray.asList(): List<Short>` Returns a [List] that wraps the original array.

`IntArray.asList(): List<Int>` Returns a [List] that wraps the original array.

`LongArray.asList(): List<Long>` Returns a [List] that wraps the original array.

`FloatArray.asList(): List<Float>` Returns a [List] that wraps the original array.

`DoubleArray.asList(): List<Double>` Returns a [List] that wraps the original array.

`BooleanArray.asList(): List<Boolean>` Returns a [List] that wraps the original array.

`CharArray.asList(): List<Char>` Returns a [List] that wraps the original array.

`Arrays.deepEquals(array1: Array<out T>, array2: Array<out T>): Boolean` Returns `true` if the two specified arrays are *deeply* equal to one another, i.e. contain the same number of the same elements in the same order. If two corresponding elements are nested arrays, they are also compared deeply. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`Arrays.contentDeepEquals(array1: Array<out T>, array2: Array<out T>): Boolean` Returns `true` if the two specified arrays are *deeply* equal to one another, i.e. contain the same number of the same elements in the same order. The specified arrays are also considered deeply equal if both are `null`. If two corresponding elements are nested arrays, they are also compared deeply. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`Arrays.contentDeepHashCode(array: Array<out T>): Int` Returns a hash code based on the contents of this array as if it is [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`Arrays.toString(array: Array<out T>): String` Returns a string representation of the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level that reference is rendered as `[...]` to prevent recursion.

`Arrays.contentDeepToString(array: Array<out T>): String` Returns a string representation of the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level that reference is rendered as `[...]` to prevent recursion.

`<T> Array<out T>?.contentDeepToString(): String`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `<T> Array<out T>.contentEquals(other: Array<out T>): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `ByteArray.contentEquals(other: ByteArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `ShortArray.contentEquals(other: ShortArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `IntArray.contentEquals(other: IntArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `LongArray.contentEquals(other: LongArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `FloatArray.contentEquals(other: FloatArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `DoubleArray.contentEquals(other: DoubleArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 Use Kotlin compiler 1.4 to avoid deprecation warning.
 Since Kotlin 1.1
 Deprecated Since Kotlin (hidden Since = 1.4)
 public expect infix fun `BooleanArray.contentEquals(other: BooleanArray): Boolean`
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it

means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")`
`@SinceKotlin("1.1")`
`@DeprecatedSinceKotlin(hiddenSince = "1.4")`
`public expect infix fun CharArray.contentEquals(other: CharArray): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun <T> Array<out T>?.contentEquals(other: Array<out T>?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun IntArray?.contentEquals(other: IntArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun LongArray?.contentEquals(other: LongArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean`
Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
`@SinceKotlin("1.4")`
`public expect infix fun CharArray?.contentHashCode(): Int`
Returns a hash code based on the contents of this array as if it is `[List]`.

`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")`
`@SinceKotlin("1.1")`
`@DeprecatedSinceKotlin(hiddenSince = "1.4")`
`public expect fun <T> Array<out T>.contentHashCode(): Int`
Returns a hash code based on the contents of this array as if it is `[List]`.

`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")`
`@SinceKotlin("1.1")`
`@DeprecatedSinceKotlin(hiddenSince = "1.4")`
`public expect fun ByteArray.contentHashCode(): Int`
Returns a hash code based on the contents of this array as if it is `[List]`.

`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")`
`@SinceKotlin("1.1")`
`@DeprecatedSinceKotlin(hiddenSince = "1.4")`
`public expect fun CharArray?.contentHashCode(): Int`
Returns a hash code based on the contents of this array as if it is `[List]`.

```

warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
ShortArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated("Use Kotlin compiler 1.4
to avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect fun IntArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if
it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
LongArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
FloatArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
DoubleArray.contentHashCode():
Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
BooleanArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
CharArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentHashCode(): Int\n\n/**\n *
Returns a hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect
fun ByteArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array
as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect fun ShortArray?.contentHashCode(): Int\n\n/**\n *
Returns a hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect
fun IntArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin("1.4")\npublic expect fun LongArray?.contentHashCode(): Int\n\n/**\n * Returns a
hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect fun
FloatArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin("1.4")\npublic expect fun DoubleArray?.contentHashCode(): Int\n\n/**\n * Returns a
hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect fun
BooleanArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents
of this array as if it is [List].\n *\n@SinceKotlin("1.4")\npublic expect fun CharArray?.contentHashCode():
Int\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect fun <T> Array<out T>.contentToString(): String\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n *\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect fun ByteArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the
specified array as if it is [List].\n *\n * \n *
@sample samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated("Use Kotlin
compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\npublic expect fun ShortArray.contentToString(): String\n\n/**\n * Returns a string representation of the
contents of the specified array as if it is [List].\n *\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated("Use Kotlin compiler 1.4 to

```



```

avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect fun IntArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the
specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
LongArray.contentToString():
String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n *
@sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler
1.4 to avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\npublic expect fun FloatArray.contentToString(): String\n\n/**\n * Returns a string representation of the
contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect fun DoubleArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the
specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
BooleanArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified
array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun
CharArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array
as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentToString(): String\n\n/**\n * Returns a
string representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\npublic expect fun ByteArray?.contentToString(): String\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun
ShortArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified
array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\npublic expect fun IntArray?.contentToString(): String\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun
LongArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified
array as if it is [List].\n
* \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\npublic expect fun FloatArray?.contentToString(): String\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun
DoubleArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified
array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\npublic expect fun BooleanArray?.contentToString(): String\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun
CharArray?.contentToString(): String\n\n/**\n * Copies this array or its subrange

```

into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n * \n @SinceKotlin("1.3")\n public expect fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>\n \n /**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n * \n @SinceKotlin("1.3")\n public expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray\n \n /**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n * \n @SinceKotlin("1.3")\n public expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray\n \n /**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n * \n @SinceKotlin("1.3")\n public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray\n \n /**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or

[endIndex] is out of range of this array indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], \n * or when that index is out of the [destination] array indices range. \n * \n * @return the [destination] array. \n * \n * @SinceKotlin("1.3")\npublic expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray\n\n\n * Copies this array or its subrange into the [destination] array and returns that array. \n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. \n * \n * @param destination the array to copy to. \n * @param destinationOffset the position in the [destination] array to copy to, 0 by default. \n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default. \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default. \n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], \n * or when that index is out of the [destination] array indices range. \n * \n * @return the [destination] array. \n * \n * @SinceKotlin("1.3")\npublic expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray\n\n\n * Copies this array or its subrange into the [destination] array and returns that array. \n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. \n * \n * @param destination the array to copy to. \n * @param destinationOffset the position in the [destination] array to copy to, 0 by default. \n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default. \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default. \n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], \n * or when that index is out of the [destination] array indices range. \n * \n * @return the [destination] array. \n * \n * @SinceKotlin("1.3")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray\n\n\n * Copies this array or its subrange into the [destination] array and returns that array. \n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. \n * \n * @param destination the array to copy to. \n * @param destinationOffset the position in the [destination] array to copy to, 0 by default. \n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default. \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default. \n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], \n * or when that index is out of the [destination] array indices range. \n * \n * @return the [destination] array. \n * \n * @SinceKotlin("1.3")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray\n\n\n * Copies this array or its subrange into the [destination] array and returns that array. \n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. \n * \n * @param destination the array to copy to. \n * @param destinationOffset the position in the [destination] array to copy to, 0 by default. \n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default. \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default. \n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], \n * or when that index is out of the [destination] array indices range. \n * \n * @return the [destination] array. \n * \n * @SinceKotlin("1.3")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):

CharArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n
 *^\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun <T> Array<T>.copyOf():
 Array<T>\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun ByteArray.copyOf():
 ByteArray\n\n/**\n * Returns
 new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun ShortArray.copyOf():
 ShortArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun IntArray.copyOf(): IntArray\n\n/**\n
 * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun LongArray.copyOf():
 LongArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun FloatArray.copyOf():
 FloatArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun DoubleArray.copyOf():
 DoubleArray\n\n/**\n * Returns new array which is a copy of the original
 array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun
 BooleanArray.copyOf(): BooleanArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n *
 @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n *^\npublic expect fun CharArray.copyOf():
 CharArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n *
 The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the
 size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size
 of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *^\npublic expect fun
 ByteArray.copyOf(newSize: Int): ByteArray\n\n/**\n * Returns new array which is a copy of the original array,
 resized to the given [newSize].\n
 * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than
 the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size
 of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *^\npublic expect fun
 ShortArray.copyOf(newSize: Int): ShortArray\n\n/**\n * Returns new array which is a copy of the original array,
 resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n
 * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If
 [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero
 values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
 *^\npublic expect fun IntArray.copyOf(newSize: Int): IntArray\n\n/**\n * Returns new array which is a copy of the
 original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values
 if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
 [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
 filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
 *^\npublic expect fun LongArray.copyOf(newSize: Int): LongArray\n\n/**\n * Returns new array which is a copy of
 the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero
 values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
 [newSize].\n * - If [newSize] is greater than the size
 of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample
 samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *^\npublic expect fun
 FloatArray.copyOf(newSize: Int): FloatArray\n\n/**\n * Returns new array which is a copy of the original array,

resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n public expect fun DoubleArray.copyOf(newSize: Int): DoubleArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `false` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n public expect fun BooleanArray.copyOf(newSize: Int): BooleanArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n public expect fun CharArray.copyOf(newSize: Int): CharArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `null` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n * \n @Suppress("NO_ACTUAL_FOR_EXPECT")\n public expect fun <T> Array<T>.copyOf(newSize: Int): Array<T?>\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @Suppress("NO_ACTUAL_FOR_EXPECT")\n public expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n public expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n public expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n public expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n


```

*^@SinceKotlin("1.3")\npublic expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int =
size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex
the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n\n^@SinceKotlin("1.3")\npublic expect fun FloatArray.fill(element: Float, fromIndex: Int = 0,
toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n
* @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
(exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
greater than [toIndex].\n\n^@SinceKotlin("1.3")\npublic expect fun DoubleArray.fill(element: Double,
fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element]
value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end
of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n\n^@SinceKotlin("1.3")\npublic
expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this
array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive)
to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n
* \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n\n
^@SinceKotlin("1.3")\npublic expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size):
Unit\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public val <T> Array<out T>.indices:
IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public
val ByteArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the
range of valid indices for the array.\n\n^@public val ShortArray.indices: IntRange\n    get() = IntRange(0,
lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public val IntArray.indices: IntRange\n
get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public val
LongArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the
array.\n\n^@public val FloatArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range
of valid indices for the array.\n\n^@public val DoubleArray.indices: IntRange\n    get() = IntRange(0,
lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public val BooleanArray.indices:
IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n\n^@public
val CharArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns `true`
if the array is empty.\n\n^@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isEmpty(): Boolean
{\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n\n^@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n\n
^@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n/**\n
* Returns `true` if the array is empty.\n\n^@kotlin.internal.InlineOnly\npublic inline fun IntArray.isEmpty():
Boolean {\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n\n
^@kotlin.internal.InlineOnly\npublic inline fun LongArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n/**\n
* Returns `true` if the array is empty.\n\n^@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isEmpty():
Boolean {\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n\n
^@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is
empty.\n\n^@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.isEmpty(): Boolean {\n    return size ==
0\n}\n\n/**\n * Returns `true` if the array is empty.\n\n^@kotlin.internal.InlineOnly\npublic inline fun
CharArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n/**\n * Returns `true` if the array is not empty.\n

```

```

*^@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isEmpty(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline
fun ByteArray.isEmpty(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array is not
empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isEmpty(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n*^@kotlin.internal.InlineOnly\npublic
inline fun IntArray.isEmpty(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array is not
empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline fun LongArray.isEmpty(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline
fun FloatArray.isEmpty(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array is not
empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isEmpty(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n*^@kotlin.internal.InlineOnly\npublic inline
fun BooleanArray.isEmpty(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val <T> Array<out
T>.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
ByteArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
ShortArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
IntArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
LongArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
FloatArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
DoubleArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
BooleanArray.lastIndex: Int\n    get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\n*^@public val
CharArray.lastIndex: Int\n    get() = size
- 1\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n*^@Suppress(\\"NO_ACTUAL_FOR_EXPECT\")\npublic expect operator fun <T> Array<T>.plus(element: T):
Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n*^@public expect operator fun ByteArray.plus(element: Byte): ByteArray\n\n/**\n * Returns an array containing all
elements of the original array and then the given [element].\n */\n*^@public expect operator fun ShortArray.plus(element: Short): ShortArray\n\n/**\n * Returns an array containing all elements of the original
array and then the given [element].\n */\n*^@public expect operator fun IntArray.plus(element: Int): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n*^@public expect
operator fun LongArray.plus(element: Long): LongArray\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n */\n*^@public expect operator fun FloatArray.plus(element: Float): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n*^@public expect
operator fun DoubleArray.plus(element: Double): DoubleArray\n\n/**\n * Returns an array containing all elements
of the original array and then the given [element].\n */\n*^@public expect operator fun BooleanArray.plus(element:
Boolean): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n */\n*^@public expect operator fun CharArray.plus(element: Char): CharArray\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n */\n*^@Suppress(\\"NO_ACTUAL_FOR_EXPECT\")\npublic expect operator fun <T> Array<T>.plus(elements:
Collection<T>): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] collection.\n */\n*^@public expect operator fun ByteArray.plus(elements:
Collection<Byte>): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] collection.\n */\n*^@public expect operator fun ShortArray.plus(elements:
Collection<Short>): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] collection.\n */\n*^@public expect operator fun IntArray.plus(elements:
Collection<Int>): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then all

```


elements of the given [elements] collection.\n */\npublic expect operator fun LongArray.plus(elements: Collection<Long>): LongArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun CharArray.plus(elements: Collection<Char>): CharArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T> Array<T>.plus(elements: Array<out T>): Array<T>\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ByteArray.plus(elements: ByteArray): ByteArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ShortArray.plus(elements: ShortArray): ShortArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun IntArray.plus(elements: IntArray): IntArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun LongArray.plus(elements: LongArray): LongArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun FloatArray.plus(elements: FloatArray): FloatArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun CharArray.plus(elements: CharArray): CharArray\n\n**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun <T> Array<T>.plusElement(element: T): Array<T>\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun IntArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun LongArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ByteArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ShortArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun DoubleArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun FloatArray.sort(): Unit\n\n**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun CharArray.sort(): Unit\n\n**\n * Sorts the array in-place according to the natural order of its elements.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n */\npublic expect fun <T : Comparable<T>> Array<out T>.sort(): Unit\n\n**\n * Sorts a range in the array in-place.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array

greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun <T : Comparable<T>> Array<out T>.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sortWith(reverseOrder(), fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun ByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun ShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun IntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun LongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun FloatArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n @SinceKotlin("1.4")\n public fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n }\n }\n /**\n * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n *\n @public expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit\n /**\n * Sorts a range in the array in-place with the given [comparator].\n

* \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n *

\n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns an array of Boolean containing all of the elements of this generic array.\n */\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n return BooleanArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Byte containing all of the elements of this generic array.\n */\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n return ByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Char containing all of the elements of this generic array.\n */\npublic fun Array<out Char>.toCharArray(): CharArray {\n return CharArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Double containing all of the elements of this generic array.\n */\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n return DoubleArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Float containing all of the elements of this generic array.\n */\npublic fun Array<out Float>.toFloatArray(): FloatArray {\n return FloatArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Int containing all of the elements of this generic array.\n */\npublic fun Array<out Int>.toIntArray(): IntArray {\n return IntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Long containing all of the elements of this generic array.\n */\npublic fun Array<out Long>.toLongArray(): LongArray {\n return LongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Short containing all of the elements of this generic array.\n */\npublic fun Array<out Short>.toShortArray(): ShortArray {\n return ShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun ByteArray.toTypedArray(): Array<Byte>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun ShortArray.toTypedArray(): Array<Short>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun IntArray.toTypedArray(): Array<Int>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun LongArray.toTypedArray(): Array<Long>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun FloatArray.toTypedArray(): Array<Float>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun DoubleArray.toTypedArray(): Array<Double>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun BooleanArray.toTypedArray(): Array<Boolean>\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have

the same key the last one gets added to the map.

`ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)}

Returns a [Map] containing the elements from the given array indexed by the key returned from [keySelector] function applied to each element. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

`Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T>` {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)}

Returns a [Map] containing the elements from the given array indexed by the key

returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Byte>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
ShortArray.associateBy(keySelector: (Short) ->
K): Map<K, Short> {\n    val capacity = mapCapacity(size).coerceAtLeast(16)\n    return
associateByTo(LinkedHashMap<K, Short>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
elements from the given array indexed by the key\n * returned from [keySelector] function applied to each
element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to
the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Int>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements
would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map
preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Long>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
FloatArray.associateBy(keySelector:
(Float) -> K): Map<K, Float> {\n    val capacity = mapCapacity(size).coerceAtLeast(16)\n    return
associateByTo(LinkedHashMap<K, Float>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
elements from the given array indexed by the key\n * returned from [keySelector] function applied to each
element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to
the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Double>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * \n * The returned map preserves the entry
iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Boolean>(capacity),

```

```

keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n *\npublic inline fun <K>
CharArray.associateBy(keySelector:
(Char) -> K): Map<K, Char> {\n    val capacity = mapCapacity(size).coerceAtLeast(16)\n    return
associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given
array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline
fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n    val
capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline
fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {\n    val
capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original
array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline
fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {\n    val
capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline
fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {\n    val
capacity = mapCapacity(size).coerceAtLeast(16)\n
return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a
[Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to
elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last
one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n
* @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n
*\npublic inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V):
Map<K, V> {\n    val capacity = mapCapacity(size).coerceAtLeast(16)\n    return
associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map]
containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of
the given array.\n
* \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n
* \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n
*\npublic inline
fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n
*\npublic inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) ->
V): Map<K, V> {\n  val capacity = mapCapacity(size).coerceAtLeast(16)\n  return
associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map]
containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of
the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets
added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n
*\npublic inline fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) ->
V): Map<K, V> {\n  val capacity = mapCapacity(size).coerceAtLeast(16)\n  return
associateByTo(LinkedHashMap<K,
V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by
[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n
*\npublic inline
fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value
pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and
value is the element itself.\n
* \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n
* \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n
*\npublic inline
fun <T, K, M : MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M
{\n  for (element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is
provided by the [keySelector] function applied to each element of the given array\n * and value is the element
itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n
*\npublic
inline fun <K, M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte)
-> K): M {\n  for (element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is
provided by the [keySelector] function applied to each element of the given array\n * and value is the element
itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n
*\npublic
inline fun <K, M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) ->
K): M {\n  for (element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is
provided by the [keySelector] function applied to each element of the given array\n * and value is the element
itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.\n * \n * @sample

```



```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M :
MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {\n for (element
in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements
would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M :
MutableMap<in K, in Long>> LongArray.associateByTo(destination:
M, keySelector: (Long) -> K): M {\n for (element in this) {\n destination.put(keySelector(element),
element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-
value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n *
and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the
last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M :
MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {\n for
(element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element
of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned
by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M :
MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M {\n
for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K,
M : MutableMap<in K, in
Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {\n for (element in
this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function
applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M :
MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K): M {\n for
(element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the
given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added
to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic
inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -
> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the

```

```

map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) ->
K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) ->
K, valueTransform: (Short) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K,
valueTransform: (Int) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with
key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the
[valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same
key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) ->
K, valueTransform: (Long) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned
by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) ->
K, valueTransform: (Float) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination:
M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {\n for (element in this) {\n
destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n *
\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n *
\n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n *\npublic
inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M,
keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {\n for (element in this) {\n

```

```

destination.put(keySelector(element), valueTransform(element))\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateTo(destination: M, transform: (Long) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {\n

```

```

for (element in this) {
    destination += transform(element)
}
return destination
}

Populates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given array.

If any of two pairs would have the same key the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given array.

If any of two pairs would have the same key the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith
@SinceKotlin("1.4")
public inline fun <K, V> Array<out K>.associateWith(valueSelector: (K) -> V): Map<K, V> {
    val result = LinkedHashMap<K, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V> ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> {
    val result = LinkedHashMap<Byte, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V> ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> {
    val result = LinkedHashMap<Short, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V> IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {
    val result = LinkedHashMap<Int, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V> LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {
    val result = LinkedHashMap<Long, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The

```

returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {n  val result = LinkedHashMap<Float,
V>(mapCapacity(size).coerceAtLeast(16))n
return associateWithTo(result, valueSelector)n}n/n/**n * Returns a [Map] where keys are elements from the
given array and values aren * produced by the [valueSelector] function applied to each element.n * n * If any two
elements are equal, the last one gets added to the map.n * n * The returned map preserves the entry iteration order
of the original array.n * n * @sample samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
DoubleArray.associateWith(valueSelector: (Double) -> V): Map<Double, V> {n  val result =
LinkedHashMap<Double, V>(mapCapacity(size).coerceAtLeast(16))n  return associateWithTo(result,
valueSelector)n}n/n/**n * Returns a [Map] where keys are elements from the given array and values aren *
produced by the [valueSelector] function applied to each element.n * n * If any two elements are equal, the last one
gets added to the map.n * n * The returned
map preserves the entry iteration order of the original array.n * n * @sample
samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V> {n  val result =
LinkedHashMap<Boolean, V>(mapCapacity(size).coerceAtLeast(16))n  return associateWithTo(result,
valueSelector)n}n/n/**n * Returns a [Map] where keys are elements from the given array and values aren *
produced by the [valueSelector] function applied to each element.n * n * If any two elements are equal, the last one
gets added to the map.n * n * The returned map preserves the entry iteration order of the original array.n * n *
@sample samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
CharArray.associateWith(valueSelector: (Char) -> V): Map<Char, V> {n  val result
= LinkedHashMap<Char, V>(mapCapacity(size).coerceAtMost(128)).coerceAtLeast(16))n  return
associateWithTo(result, valueSelector)n}n/n/**n * Populates and returns the [destination] mutable map with key-
value pairs for each element of the given array,n * where key is the element itself and value is provided by the
[valueSelector] function applied to that key.n * n * If any two elements are equal, the last one overwrites the
former value in the map.n * n * @sample samples.collections.Collections.Transformations.associateWithTo
*/n@SinceKotlin("1.4")npublic inline fun <K, V, M : MutableMap<in K, in V>> Array<out
K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {n  for (element in this) {n
destination.put(element, valueSelector(element))n  }n  return destinationn}n/n/**n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,n * where key is the element
itself and value is provided by
the [valueSelector] function applied to that key.n * n * If any two elements are equal, the last one overwrites the
former value in the map.n * n * @sample samples.collections.Collections.Transformations.associateWithTo
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V, M : MutableMap<in Byte, in V>>
ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M {n  for (element in this) {n
destination.put(element, valueSelector(element))n  }n  return destinationn}n/n/**n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.n * n * If any two elements are
equal, the last one overwrites the former value in the map.n * n * @sample
samples.collections.Collections.Transformations.associateWithTo
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic
inline fun <V, M : MutableMap<in Short, in V>> ShortArray.associateWithTo(destination: M, valueSelector:
(Short) -> V): M {n  for (element in this) {n  destination.put(element, valueSelector(element))n  }n  return

```

```

destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Int, in V>>
IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination]
mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value
is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one
overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n
* @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {\n  for (element
in this) {\n    destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n *
where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If
any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Boolean, in
V>> BooleanArray.associateWithTo(destination: M, valueSelector: (Boolean) -> V): M {\n  for (element in this)
{\n    destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the
element itself and value is provided
by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites
the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Char, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Appends all elements to
the given [destination] collection.\n * \n * @public fun <T, C : MutableCollection<in T>> Array<out
T>.toCollection(destination: C): C {\n  for (item in this) {\n    destination.add(item)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n * \n * @public fun <C :
MutableCollection<in Byte>> ByteArray.toCollection(destination: C): C {\n  for (item in this) {\n
destination.add(item)\n

```

```

    } \n return destination \n } \n \n /** \n * Appends all elements to the given [destination] collection. \n * \n public fun
<C : MutableCollection<in Short>> ShortArray.toCollection(destination: C): C { \n for (item in this) { \n
destination.add(item) \n } \n return destination \n } \n \n /** \n * Appends all elements to the given [destination]
collection. \n * \n public fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C { \n for
(item in this) { \n destination.add(item) \n } \n return destination \n } \n \n /** \n * Appends all elements to the
given [destination] collection. \n * \n public fun <C : MutableCollection<in Long>>
LongArray.toCollection(destination: C): C { \n for (item in this) { \n destination.add(item) \n } \n return
destination \n } \n \n /** \n * Appends all elements to the given [destination] collection. \n * \n public fun <C :
MutableCollection<in Float>> FloatArray.toCollection(destination: C): C { \n for (item
in this) { \n destination.add(item) \n } \n return destination \n } \n \n /** \n * Appends all elements to the given
[destination] collection. \n * \n public fun <C : MutableCollection<in Double>>
DoubleArray.toCollection(destination: C): C { \n for (item in this) { \n destination.add(item) \n } \n return
destination \n } \n \n /** \n * Appends all elements to the given [destination] collection. \n * \n public fun <C :
MutableCollection<in Boolean>> BooleanArray.toCollection(destination: C): C { \n for (item in this) { \n
destination.add(item) \n } \n return destination \n } \n \n /** \n * Appends all elements to the given [destination]
collection. \n * \n public fun <C : MutableCollection<in Char>> CharArray.toCollection(destination: C): C { \n for
(item in this) { \n destination.add(item) \n } \n return destination \n } \n \n /** \n * Returns a new [HashSet] of all
elements. \n * \n public fun <T> Array<out T>.toHashSet(): HashSet<T> { \n return
toCollection(HashSet<T>(mapCapacity(size))) \n } \n \n /** \n
* Returns a new [HashSet] of all elements. \n * \n public fun ByteArray.toHashSet(): HashSet<Byte> { \n return
toCollection(HashSet<Byte>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun ShortArray.toHashSet(): HashSet<Short> { \n return
toCollection(HashSet<Short>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun IntArray.toHashSet(): HashSet<Int> { \n return
toCollection(HashSet<Int>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun LongArray.toHashSet(): HashSet<Long> { \n return
toCollection(HashSet<Long>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun FloatArray.toHashSet(): HashSet<Float> { \n return
toCollection(HashSet<Float>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun DoubleArray.toHashSet(): HashSet<Double> { \n
return toCollection(HashSet<Double>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all
elements. \n * \n public fun BooleanArray.toHashSet(): HashSet<Boolean> { \n return
toCollection(HashSet<Boolean>(mapCapacity(size))) \n } \n \n /** \n * Returns a new [HashSet] of all elements. \n
* \n public fun CharArray.toHashSet(): HashSet<Char> { \n return
toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128)))) \n } \n \n /** \n * Returns a [List] containing all
elements. \n * \n public fun <T> Array<out T>.toList(): List<T> { \n return when (size) { \n 0 -> emptyList() \n
1 -> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n /** \n * Returns a [List] containing all
elements. \n * \n public fun ByteArray.toList(): List<Byte> { \n return when (size) { \n 0 -> emptyList() \n 1
-> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n /** \n * Returns a [List] containing all elements. \n
* \n public fun ShortArray.toList(): List<Short> { \n
return when (size) { \n 0 -> emptyList() \n 1 -> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n
/** \n * Returns a [List] containing all elements. \n * \n public fun IntArray.toList(): List<Int> { \n return
when (size) { \n 0 -> emptyList() \n 1 -> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n /** \n
Returns a [List] containing all elements. \n * \n public fun LongArray.toList(): List<Long> { \n return when (size)
{ \n 0 -> emptyList() \n 1 -> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n /** \n * Returns a
[List] containing all elements. \n * \n public fun FloatArray.toList(): List<Float> { \n return when (size) { \n 0 ->
emptyList() \n 1 -> listOf(this[0]) \n else -> this.toMutableList() \n } \n } \n \n /** \n * Returns a [List]
containing all elements. \n * \n public fun DoubleArray.toList(): List<Double> { \n return when (size) { \n 0 ->

```

```

emptyList()\n    1
-> listOf(this[0])\n    else -> this.toMutableList()\n    }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun BooleanArray.toList(): List<Boolean> {\n    return when (size) {\n        0 -> emptyList()\n        1 -> listOf(this[0])\n        else -> this.toMutableList()\n    }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun CharArray.toList(): List<Char> {\n    return when (size) {\n        0 -> emptyList()\n        1 -> listOf(this[0])\n        else -> this.toMutableList()\n    }\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun <T> Array<out T>.toMutableList(): MutableList<T> {\n    return ArrayList(this.asCollection())\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun ByteArray.toMutableList(): MutableList<Byte> {\n    val list = ArrayList<Byte>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun ShortArray.toMutableList(): MutableList<Short> {\n    val list = ArrayList<Short>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun IntArray.toMutableList(): MutableList<Int> {\n    val list = ArrayList<Int>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun LongArray.toMutableList(): MutableList<Long> {\n    val list = ArrayList<Long>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun FloatArray.toMutableList(): MutableList<Float> {\n    val list = ArrayList<Float>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun DoubleArray.toMutableList(): MutableList<Double> {\n    val list = ArrayList<Double>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun BooleanArray.toMutableList(): MutableList<Boolean> {\n    val list = ArrayList<Boolean>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun CharArray.toMutableList(): MutableList<Char> {\n    val list = ArrayList<Char>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun <T> Array<out T>.toSet(): Set<T> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ByteArray.toSet(): Set<Byte> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ShortArray.toSet(): Set<Short> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun IntArray.toSet(): Set<Int> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun LongArray.toSet(): Set<Long> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun FloatArray.toSet(): Set<Float> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun DoubleArray.toSet(): Set<Double> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n    }\n}\n

```



```

}
}

Returns a [Set] of all elements.
The returned set preserves the element iteration order of the original array.
public fun BooleanArray.toSet(): Set<Boolean> {
    return when (size) {
        0 -> emptySet()
        1 -> setOf(this[0])
        else -> toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))
    }
}

Returns a [Set] of all elements.
The returned set preserves the element iteration order of the original array.
public fun CharArray.toSet(): Set<Char> {
    return when (size) {
        0 -> emptySet()
        1 -> setOf(this[0])
        else -> toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))
    }
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <T, R> Array<out T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> ShortArray.flatMap(transform: (Short) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> IntArray.flatMap(transform: (Int) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> LongArray.flatMap(transform: (Long) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> CharArray.flatMap(transform: (Char) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
@sample samples.collections.Collections.Transformations.flatMap
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapSequence")
public inline fun <T, R> Array<out T>.flatMap(transform: (T) -> Sequence<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
@sample samples.collections.Collections.Transformations.flatMapIndexed

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun
<T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n
* and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the
original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded

```

```

from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.flatMapIndexed(transform: (index: Int, Char)
-> Iterable<R>): List<R> {\n    return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a
single list of all elements yielded from results of [transform] function being invoked on each element\n * and its
index in the original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c
inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index:
Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,
element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Byte) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val
list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends
all elements yielded from results of [transform] function being invoked on each element\n * and its index in the
original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Short) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,
element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c
inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Int) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Long)

```



```

destination\n}\n\n/**\n * Appends all elements yielded from results
of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic
inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C
{\n  for (element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {\n  for (element in this) {\n    val
list = transform(element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {\n  for (element in this) {\n    val
list = transform(element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapTo(destination: C,
transform: (Double) -> Iterable<R>): C {\n  for (element in this) {\n    val list = transform(element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline
fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) ->
Iterable<R>): C {\n  for (element in this) {\n    val list = transform(element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline
fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C
{\n  for (element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n */\npublic inline fun <T, R, C :
MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Groups elements of the original array by the
key returned by the given [keySelector] function\n * applied to each element and returns a map where each group
key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order
of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K> Array<out
T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic
inline fun <K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map

```

where each group key is associated

```
with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)\n}\n\n/**\n * Groups elements of the original
array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where
each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry
iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K>
LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K>
FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<Float>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K>
DoubleArray.groupBy(keySelector:
(Double) -> K): Map<K, List<Double>> {\n  return groupByTo(LinkedHashMap<K, MutableList<Double>>(),
keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun
<K> BooleanArray.groupBy(keySelector: (Boolean) -> K): Map<K, List<Boolean>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<Boolean>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration
order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K>
CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied
to each element of the original array\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The
returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\npublic inline fun <T, K, V>
Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the
key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key
is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the
keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\npublic inline fun <K, V>
ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
```

returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned

map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample

```

samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>> {\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>> {\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>> {\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map
where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
DoubleArray.groupBy(keySelector:
(Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform]
function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding
values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n
* \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun
<K, V> BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>>
{\n
return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n *
Groups values returned by the [valueTransform] function applied to each element of the original array\n
* by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration
order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n *\npublic inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n
return

```

groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K, M : MutableMap<in K, MutableList<T>>>

```
Array<out T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<T>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Byte>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Byte>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Short>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Short>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Int>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Int>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Long>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Long>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Float>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Float>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Double>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```

```
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<Double>() }\n        list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
```



```

samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,
MutableList<Boolean>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {\n for
(element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<Boolean>() }\n list.add(element)\n
}\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given
[keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with
a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,
MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in
this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Char>() }\n
list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to
the element\n * and puts to the [destination] map each group key associated with a list of corresponding
values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic
inline fun <K, V, M : MutableMap<in K, MutableList<V>>>
ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {\n for
(element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key
associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K,
valueTransform: (Short) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key)
{ ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key
returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group
key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K,
valueTransform: (Int) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each
element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n *
and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return
The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K,
valueTransform: (Long) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =

```


`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> IntArray.map(transform: (Int) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample

`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> LongArray.map(transform: (Long) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample

`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> FloatArray.map(transform: (Float) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample

`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> DoubleArray.map(transform: (Double) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample

`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> BooleanArray.map(transform: (Boolean) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample

`samples.collections.Collections.Transformations.map`\n *\npublic inline fun <R> CharArray.map(transform: (Char) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <T, R> Array<out T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R>

`ByteArray.mapIndexed`(transform: (index: Int, Byte) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R> ShortArray.mapIndexed(transform: (index: Int, Short) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R> IntArray.mapIndexed(transform: (index: Int, Int) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\npublic inline fun <R> DoubleArray.mapIndexed(transform: (index: Int, Double) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list

containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> BooleanArray.mapIndexed(transform: (index: Int, Boolean) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> CharArray.mapIndexed(transform: (index: Int, Char) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any> Array<out T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n return mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends only the non-null results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Array<out T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic

```

inline fun <R, C : MutableCollection<in R>> FloatArray.mapIndexedTo(destination: C, transform: (index: Int,
Float) -> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n
* and appends the results to the given [destination].\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline
fun <R, C : MutableCollection<in R>> DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double)
-> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return
destination\n}\n\n/**\n * Applies the given [transform] function\n * to each element and its index in the original array\n
* and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n
* and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.mapIndexedTo(destination:
C, transform: (index: Int, Boolean) -> R): C {\n  var index = 0\n  for (item in this)\n
destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given [transform]
function to each element and its index in the original array\n * and appends the results to the given [destination].\n
* @param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>>
CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Returns
a list containing only the non-null results of applying the given [transform] function\n * to each element in the
original array.\n * @sample samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline
fun <T, R : Any> Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {\n  return
mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element in
the original array\n * and appends only the non-null results to the given [destination].\n */\npublic inline fun <T, R :
Any, C : MutableCollection<in R>> Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n
forEach { element -> transform(element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies
the given [transform] function to each element of the original array\n * and appends the results to the given
[destination].\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapTo(destination: C,
transform:
(T) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n *
Applies the given [transform] function to each element of the original array\n * and appends the results to the given
[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapTo(destination: C,
transform: (Byte) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and
appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
ShortArray.mapTo(destination: C, transform: (Short) -> R): C {\n  for (item in this)\n
destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {\n  for (item in this)\n
destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {\n  for (item in this)\n
destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to
each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R,
C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {\n  for (item in
this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results

```

```

to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {\n  for (item in this)\n
destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C {\n  for (item in
this)\n  destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic
inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C {\n  for
(item in this)\n  destination.add(transform(item))\n  return destination\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and
the element itself.\n */\npublic fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {\n  return
IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n
* into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun
ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n *
Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the
index of that element and the element itself.\n */\npublic fun ShortArray.withIndex():
Iterable<IndexedValue<Short>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n */\npublic fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {\n  return IndexingIterable {
iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue]
containing the index of that element and the element itself.\n */\npublic fun LongArray.withIndex():
Iterable<IndexedValue<Long>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n */\npublic fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {\n  return
IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n
* into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun
DoubleArray.withIndex(): Iterable<IndexedValue<Double>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n *
Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the
index of that element and the element itself.\n */\npublic fun BooleanArray.withIndex():
Iterable<IndexedValue<Boolean>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy
[Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that
element and the element itself.\n */\npublic fun CharArray.withIndex(): Iterable<IndexedValue<Char>> {\n  return
IndexingIterable { iterator() }\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n *
\n * Among equal elements of the given array, only the first one will be present in the resulting list.\n * The
elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T> Array<out
T>.distinct(): List<T>
{\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the
given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n *
\n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun
ByteArray.distinct(): List<Byte> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing
only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun ShortArray.distinct():
List<Short> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements
from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source
array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n

```

```

*^public fun IntArray.distinct(): List<Int> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public fun LongArray.distinct(): List<Long> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public fun FloatArray.distinct(): List<Float> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public fun DoubleArray.distinct(): List<Double> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public fun BooleanArray.distinct(): List<Boolean> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public fun CharArray.distinct(): List<Char> {\n  return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * Among elements of the given array with equal keys, only the first one will be present in the resulting list.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public inline fun <T, K> Array<out T>.distinctBy(selector: (T) -> K): List<T> {\n  val set = HashSet<K>()\n  val list = ArrayList<T>()\n  for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n      list.add(e)\n  }\n  return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public inline fun <K> ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {\n  val set = HashSet<K>()\n  val list = ArrayList<Byte>()\n  for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n      list.add(e)\n  }\n  return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public inline fun <K> ShortArray.distinctBy(selector: (Short) -> K): List<Short> {\n  val set = HashSet<K>()\n  val list = ArrayList<Short>()\n  for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n      list.add(e)\n  }\n  return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public inline fun <K> IntArray.distinctBy(selector: (Int) -> K): List<Int> {\n  val set = HashSet<K>()\n  val list = ArrayList<Int>()\n  for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n      list.add(e)\n  }\n  return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^public inline fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {\n  val set = HashSet<K>()\n  val list = ArrayList<Long>()\n
```

```

    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return
list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by
the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the
source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic
inline fun <K> FloatArray.distinctBy(selector: (Float) -> K): List<Float> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Float>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K>
DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Double>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n
    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K>
BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Boolean>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n
    }\n    return list\n}\n\n/**\n * Returns
a list containing only elements from the given array\n * having distinct keys returned by the given [selector]
function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n *
@sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K>
CharArray.distinctBy(selector: (Char) -> K): List<Char> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Char>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n
    }\n    return list\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun <T>
Array<out T>.intersect(other: Iterable<T>): Set<T> {\n    val set
= this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that
are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n */\npublic infix fun ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {\n    val set =
this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are
contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n */\npublic infix fun ShortArray.intersect(other: Iterable<Short>): Set<Short> {\n    val set
= this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the
specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To
get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix
fun IntArray.intersect(other: Iterable<Int>): Set<Int> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun
LongArray.intersect(other: Iterable<Long>): Set<Long> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns
a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned
set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are

```


contained at least in one of these collections use [union].

```
public infix fun FloatArray.intersect(other:
Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}
```

Returns a set containing all elements that are contained by both this array and the specified collection. The returned set preserves the element iteration order of the original array. To get a set containing all elements that are contained at least in one of these collections use [union].

```
public infix fun DoubleArray.intersect(other:
Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return
set
}
```

Returns a set containing all elements that are contained by both this array and the specified collection. The returned set preserves the element iteration order of the original array. To get a set containing all elements that are contained at least in one of these collections use [union].

```
public infix fun BooleanArray.intersect(other: Iterable<Boolean>):
Set<Boolean> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}
```

Returns a set containing all elements that are contained by both this array and the specified collection. The returned set preserves the element iteration order of the original array. To get a set containing all elements that are contained at least in one of these collections use [union].

```
public infix fun CharArray.intersect(other:
Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun <T> Array<out T>.subtract(other: Iterable<T>): Set<T> {
    val set =
this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun ByteArray.subtract(other: Iterable<Byte>):
Set<Byte> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun IntArray.subtract(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun LongArray.subtract(other: Iterable<Long>): Set<Long> {
    val set =
this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun FloatArray.subtract(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun DoubleArray.subtract(other: Iterable<Double>): Set<Double> {
    val set =
this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun BooleanArray.subtract(other: Iterable<Boolean>):
Set<Boolean> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun CharArray.subtract(other: Iterable<Char>): Set<Char> {
    val set =
this.toMutableSet()
    set.removeAll(other)
    return set
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original

```

array.\n *^\npublic fun <T> Array<out T>.toMutableSet(): MutableSet<T> {\n  return
toCollection(LinkedHashSet<T>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original
array.\n *^\npublic fun ByteArray.toMutableSet(): MutableSet<Byte> {\n  return
toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given
array.\n * \n * The returned set preserves the element iteration order of the original array.\n *^\npublic fun
ShortArray.toMutableSet(): MutableSet<Short> {\n  return
toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original
array.\n *^\npublic fun IntArray.toMutableSet(): MutableSet<Int> {\n  return
toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original
array.\n *^\npublic fun LongArray.toMutableSet(): MutableSet<Long> {\n  return
toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given array.\n * \n * The returned set preserves the element iteration order
of the original array.\n *^\npublic fun FloatArray.toMutableSet(): MutableSet<Float> {\n  return
toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all
distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original
array.\n *^\npublic fun DoubleArray.toMutableSet(): MutableSet<Double> {\n  return
toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing
all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the
original array.\n *^\npublic fun BooleanArray.toMutableSet(): MutableSet<Boolean> {\n  return
toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing
all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the
original array.\n *^\npublic fun CharArray.toMutableSet():
MutableSet<Char> {\n  return
toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n}\n\n/**\n * Returns a set containing
all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the
original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of
the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use
[intersect].\n *^\npublic infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {\n  val set =
this.toMutableSet()\n  set.addAll(other)\n  return set\n}\n\n/**\n * Returns a set containing all distinct elements
from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those
elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n *
\n * To get a set containing
all elements that are contained in both collections use [intersect].\n *^\npublic infix fun ByteArray.union(other:
Iterable<Byte>): Set<Byte> {\n  val set = this.toMutableSet()\n  set.addAll(other)\n  return set\n}\n\n/**\n *
Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element
iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the
end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both
collections use [intersect].\n *^\npublic infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {\n  val set =
this.toMutableSet()\n  set.addAll(other)\n  return set\n}\n\n/**\n * Returns a set containing all distinct elements
from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those
elements of the [other] collection that are
unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements
that are contained in both collections use [intersect].\n *^\npublic infix fun IntArray.union(other: Iterable<Int>):
Set<Int> {\n  val set = this.toMutableSet()\n  set.addAll(other)\n  return set\n}\n\n/**\n * Returns a set

```

containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n * \n\npublic infix fun LongArray.union(other: Iterable<Long>): Set<Long> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n * \n\npublic infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n * \n\npublic infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n * \n\npublic infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n * \n\npublic infix fun CharArray.union(other: Iterable<Char>): Set<Char> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample

```

samples.collections.Collections.Aggregates.all\n *^\npublic inline fun CharArray.all(predicate: (Char) -> Boolean):
Boolean {\n  for (element in this) if (!predicate(element)) return false\n  return true\n}\n\n/**\n * Returns `true` if
array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n *^\npublic fun
<T> Array<out T>.any(): Boolean {\n  return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *^\npublic fun ByteArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *^\npublic fun ShortArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *^\npublic fun IntArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *^\npublic fun LongArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *^\npublic fun FloatArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns
`true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
*^\npublic fun DoubleArray.any(): Boolean {\n  return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least
one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n *^\npublic fun
BooleanArray.any(): Boolean {\n  return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n
* \n * @sample samples.collections.Collections.Aggregates.any\n *^\npublic fun CharArray.any(): Boolean {\n
return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic inline fun <T> Array<out
T>.any(predicate: (T) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*^\npublic inline fun ByteArray.any(predicate: (Byte) -> Boolean): Boolean {\n  for (element in this) if
(predicate(element)) return true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the
given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic
inline fun ShortArray.any(predicate: (Short) -> Boolean): Boolean {\n  for (element in this) if (predicate(element))
return true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic inline fun
IntArray.any(predicate: (Int) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n
return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*^\npublic inline fun LongArray.any(predicate: (Long) -> Boolean): Boolean {\n  for (element in this) if
(predicate(element)) return true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the
given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic
inline fun FloatArray.any(predicate: (Float) -> Boolean): Boolean {\n  for (element in this) if (predicate(element))
return true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic inline fun
DoubleArray.any(predicate: (Double) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return
true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic
inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {\n  for (element in this) if
(predicate(element)) return true\n  return false\n}\n\n/**\n * Returns `true` if at least one element matches the
given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\npublic
inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element))
return true\n  return false\n}\n\n/**\n * Returns the number of elements in this array.\n

```

```

*  

@kotlin.internal.InlineOnly  

public inline fun <T> Array<out T>.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun  

ByteArray.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun ShortArray.count(): Int {  

    return size  

}  

Returns the  

number of elements in this  

array.  

*  

@kotlin.internal.InlineOnly  

public inline fun IntArray.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun  

LongArray.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun FloatArray.count(): Int {  

    return size  

}  

Returns the  

number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun DoubleArray.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public  

inline fun BooleanArray.count(): Int {  

    return size  

}  

Returns the number of elements in this array.  

*  

@kotlin.internal.InlineOnly  

public inline fun CharArray.count(): Int {  

    return size  

}  

Returns the  

number of elements matching the given [predicate].  

*  

public inline fun <T> Array<out  

T>.count(predicate: (T) -> Boolean): Int {  

    var count = 0  

    for (element in this) if (predicate(element))  

        ++count  

    return count  

}  

Returns the number of elements matching the given [predicate].  

*  

public  

inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {  

    var count = 0  

    for (element in this) if  

    (predicate(element)) ++count  

    return count  

}  

Returns the number of elements matching the given  

[predicate].  

*  

public inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {  

    var count = 0  

    for  

    (element in this) if (predicate(element)) ++count  

    return count  

}  

Returns the number of elements  

matching the given [predicate].  

*  

public inline fun IntArray.count(predicate: (Int) -> Boolean): Int {  

    var  

    count = 0  

    for (element in this) if (predicate(element)) ++count  

    return count  

}  

Returns the  

number of elements matching the given [predicate].  

*  

public inline fun LongArray.count(predicate:  

(Long) -> Boolean): Int {  

    var count = 0  

    for (element in this) if (predicate(element)) ++count  

    return  

    count  

}  

Returns the number of elements matching the given [predicate].  

*  

public inline fun  

FloatArray.count(predicate: (Float) -> Boolean): Int {  

    var count = 0  

    for (element in this) if  

    (predicate(element)) ++count  

    return count  

}  

Returns the number of elements matching the given  

[predicate].  

*  

public inline fun DoubleArray.count(predicate: (Double) -> Boolean): Int {  

    var count = 0  

    for  

    (element in this) if (predicate(element)) ++count  

    return count  

}  

Returns the number of elements  

matching the given [predicate].  

*  

public inline fun BooleanArray.count(predicate: (Boolean) -> Boolean): Int  

{  

    var count = 0  

    for (element in this) if (predicate(element)) ++count  

    return count  

}  

Returns  

the number of elements matching the given [predicate].  

*  

public inline fun  

CharArray.count(predicate: (Char) -> Boolean): Int {  

    var count = 0  

    for (element in this) if  

    (predicate(element)) ++count  

    return count  

}  

Accumulates value starting with [initial] value and  

applying [operation] from left to right  

* to current accumulator value and each element.  

* Returns the  

specified [initial] value if the array is empty.  

* @param [operation] function that takes current accumulator  

value and an element, and calculates the next accumulator value.  

*  

public inline fun <T, R> Array<out  

T>.fold(initial: R, operation: (acc: R, T) -> R): R {  

    var accumulator = initial  

    for (element in this)  

    accumulator = operation(accumulator, element)  

    return accumulator  

}  

Accumulates value starting  

with [initial] value and applying [operation] from left to right  

* to current accumulator value and each element.  

* Returns the specified [initial] value if the array is empty.  

* @param [operation] function that  

takes current accumulator value and an element, and calculates the next accumulator value.  

*  

public inline fun  

<R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {  

    var accumulator = initial  

    for (element in  

    this) accumulator = operation(accumulator, element)  

    return accumulator  

}  

Accumulates value  

starting with [initial] value and applying [operation] from left to right  

* to current accumulator value and each  

element.  

* Returns the specified [initial] value if the array is empty.  

* @param [operation] function that  

takes current accumulator value and an element, and calculates the next accumulator value.  

*  

public inline fun  

<R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {  

    var accumulator = initial  

    for (element in

```

```

this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n
 * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the
next accumulator value.\n */\npublic inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {\n
var accumulator = initial\n  for (element in this) accumulator = operation(accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the
next accumulator value.\n */\npublic inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R
{\n  var accumulator = initial\n  for (element in this) accumulator = operation(accumulator,
element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified
[initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.fold(initial: R,
operation: (acc: R, Float) -> R): R {\n  var accumulator = initial\n  for (element in this) accumulator =
operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns
the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n */\npublic
inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {\n  var accumulator = initial\n
for (element in this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator
value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n */\npublic inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n  var
accumulator = initial\n  for (element in this) accumulator = operation(accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial]
value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.fold(initial: R, operation:
(acc: R, Char) -> R): R {\n  var accumulator = initial\n  for (element in this) accumulator =
operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from left to right\n * to current accumulator value and each element with its index in
the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int,
acc: R, T) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element
in this) accumulator = operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and
each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n
 * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldIndexed(initial:
R, operation: (index: Int, acc: R, Byte) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element in
this) accumulator = operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and
each element with its index in the original array.\n * \n * Returns

```

the specified [initial] value if the array is empty.

```

    * @param [operation] function that takes the index of an
    element, current accumulator value
    * and the element itself, and calculates the next accumulator value.
    */
    public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
        var index = 0
        var accumulator = initial
        for (element in this) accumulator = operation(index++, accumulator, element)
        return accumulator
    }
    * Accumulates value starting with [initial] value and applying
    [operation] from left to right
    * to current accumulator value and each element with its index in the original array.
    * Returns the specified [initial] value if the array is empty.
    * @param [operation] function that takes the index of an element, current accumulator value
    * and the element itself, and calculates the next accumulator
    value.
    */
    public inline fun <R> IntArray.foldIndexed(initial: R, operation:
    (index: Int, acc: R, Int) -> R): R {
        var index = 0
        var accumulator = initial
        for (element in this)
        accumulator = operation(index++, accumulator, element)
        return accumulator
    }
    * Accumulates value
    starting with [initial] value and applying [operation] from left to right
    * to current accumulator value and each element with its index in the original array.
    * Returns the specified [initial] value if the array is empty.
    * @param [operation] function that takes the index of an element, current accumulator value
    * and the element itself, and calculates the next accumulator value.
    */
    public inline fun <R> LongArray.foldIndexed(initial: R,
    operation: (index: Int, acc: R, Long) -> R): R {
        var index = 0
        var accumulator = initial
        for (element in
        this) accumulator = operation(index++, accumulator, element)
        return accumulator
    }
    * Accumulates
    value starting with [initial] value and applying [operation] from left to right
    * to current accumulator value and each element with its index in the original array.
    * Returns the specified
    [initial] value if the array is empty.
    * @param [operation] function that takes the index of an element, current
    accumulator value
    * and the element itself, and calculates the next accumulator value.
    */
    public inline fun <R>
    FloatArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): R {
        var index = 0
        var
        accumulator = initial
        for (element in this) accumulator = operation(index++, accumulator, element)
        return
        accumulator
    }
    * Accumulates value starting with [initial] value and applying [operation] from left to
    right
    * to current accumulator value and each element with its index in the original array.
    * Returns the
    specified [initial] value if the array is empty.
    * @param [operation] function that takes the index of an
    element, current accumulator value
    * and the element itself, and calculates
    the next accumulator value.
    */
    public inline fun <R> DoubleArray.foldIndexed(initial: R, operation: (index: Int,
    acc: R, Double) -> R): R {
        var index = 0
        var accumulator = initial
        for (element in this) accumulator =
        operation(index++, accumulator, element)
        return accumulator
    }
    * Accumulates value starting with
    [initial] value and applying [operation] from left to right
    * to current accumulator value and each element with its
    index in the original array.
    * Returns the specified [initial] value if the array is empty.
    * @param
    [operation] function that takes the index of an element, current accumulator value
    * and the element itself, and
    calculates the next accumulator value.
    */
    public inline fun <R> BooleanArray.foldIndexed(initial: R, operation:
    (index: Int, acc: R, Boolean) -> R): R {
        var index = 0
        var accumulator = initial
        for (element in this)
        accumulator = operation(index++, accumulator, element)
        return accumulator
    }
    * Accumulates value starting with [initial] value and applying [operation] from left to right
    * to current
    accumulator value and each element with its index in the original array.
    * Returns the specified [initial] value
    if the array is empty.
    * @param [operation] function that takes the index of an element, current accumulator
    value
    * and the element itself, and calculates the next accumulator value.
    */
    public inline fun <R>
    CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {
        var index = 0
        var
        accumulator = initial
        for (element in this) accumulator = operation(index++, accumulator, element)
        return
        accumulator
    }
    * Accumulates value starting with [initial] value and applying [operation] from right to
    left
    * to each element and current accumulator value.
    * Returns the specified [initial] value if the array is
    empty.
    * @param [operation] function that takes an element and
    current accumulator value, and calculates the next accumulator value.
    */
    public inline fun <T, R> Array<out
    T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {
        var index = lastIndex
        var accumulator = initial
        while (index >= 0) {
            accumulator = operation(get(index--), accumulator)
        }
        return
    }

```

```

accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R):
R {\n    var index = lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n        accumulator =
operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from right to left\n * to each element and current
accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes an element and current accumulator value, and calculates the next accumulator value.\n
*/\npublic inline fun <R> ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R {\n    var index =
lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n        accumulator = operation(get(index--),
accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified
[initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R>
IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R {\n    var index = lastIndex\n    var accumulator =
initial\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n */\npublic inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R):
R {\n    var index = lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n        accumulator =
operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each element
and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value, and calculates the next accumulator
value.\n */\npublic inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {\n    var index
= lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n        accumulator = operation(get(index--),
accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified
[initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R>
DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {\n    var index = lastIndex\n
var accumulator = initial\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n
}\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from
right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the
array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and
calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.foldRight(initial: R, operation:
(Boolean, acc: R) -> R): R {\n    var index = lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates
the next accumulator value.\n */\npublic inline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) ->
R): R {\n    var index = lastIndex\n    var accumulator = initial\n    while (index >= 0) {\n        accumulator =
operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param

```



```

[operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <T, R> Array<out T>.foldRightIndexed(initial: R,
operation: (index: Int, T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying
[operation] from right to left to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from
right to left to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from right to left
to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the
element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from right to left
to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from
right to left to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from right to
left to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element itself
and current accumulator value, and calculates the next accumulator value.
public inline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (index: Int, Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
* Accumulates value starting with [initial] value and applying [operation] from right to left
to each element with its index in the original array and current accumulator value.
* Returns the specified [initial] value if the array is empty.
* @param [operation] function that takes the index of an element, the element

```

```

    itself\n * and current accumulator value, and calculates the next accumulator value.\n *\npublic inline fun <R>
BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {\n  var index =
lastIndex\n  var accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from right to left\n * to each element with its index in the original array and current
accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the
next accumulator value.\n */\npublic inline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int,
Char, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator
= initial\n  while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n
}\n  return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun <T>
Array<out T>.forEach(action: (T) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n * Performs
the given [action] on each element.\n */\npublic inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {\n  for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun
ShortArray.forEach(action: (Short) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n * Performs
the given [action] on each element.\n */\npublic inline fun IntArray.forEach(action: (Int) -> Unit): Unit {\n  for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun
LongArray.forEach(action: (Long) -> Unit): Unit {\n  for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun
FloatArray.forEach(action: (Float) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n * Performs
the given [action] on each element.\n */\npublic inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {\n
for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline
fun BooleanArray.forEach(action: (Boolean) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element.\n */\npublic inline fun CharArray.forEach(action: (Char) -> Unit):
Unit {\n  for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing
sequential index with the element.\n * @param [action] function that takes the index of an element and the element
itself\n * and performs the action on the element.\n */\npublic
inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n  var index = 0\n  for (item
in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and
performs the action on the element.\n */\npublic inline fun ByteArray.forEachIndexed(action: (index: Int, Byte) ->
Unit): Unit {\n  var index = 0\n  for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action]
on each element, providing sequential index with the element.\n * @param [action] function that takes the index of
an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun
ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit): Unit {\n  var index = 0\n  for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing
sequential index with the element.\n * @param [action] function that takes the index of an element and the element
itself\n * and performs the action on the element.\n */\npublic inline fun IntArray.forEachIndexed(action: (index:
Int, Int) -> Unit): Unit {\n  var index = 0\n  for (item in this) action(index++, item)\n}\n\n/**\n * Performs
the given [action] on each element, providing sequential index with the element.\n * @param [action] function that
takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline
fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit): Unit {\n  var index = 0\n  for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun FloatArray.forEachIndexed(action:
(index: Int, Float) -> Unit): Unit {\n  var index = 0\n  for (item in this) action(index++, item)\n}\n\n/**\n *
Performs the given [action] on each element, providing sequential index with the element.\n * @param [action]
function that takes the index of an element and the element itself\n * and performs the action on the element.\n

```

```

*/\npublic inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit): Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n\npublic inline fun BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n\npublic inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Double>.max(): Double? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Float>.max(): Float? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Array<out T>.max(): T? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun ByteArray.max(): Byte? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun ShortArray.max(): Short? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun IntArray.max(): Int? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun LongArray.max(): Long? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun FloatArray.max(): Float? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun DoubleArray.max(): Double? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun CharArray.max(): Char? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T? {\n    return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? {\n    return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short? {\n    return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int?

```

```

{\n  return maxByOrNull(selector)\n}\n\n@Deprecated(\\"Use maxByOrNull
instead.\", ReplaceWith(\\"this.maxByOrNull(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = \\"1.4\",
errorSince = \\"1.5\", hiddenSince = \\"1.6\")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector:
(Long) -> R): Long? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated(\\"Use maxByOrNull instead.\",
ReplaceWith(\\"this.maxByOrNull(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = \\"1.4\", errorSince =
\\"1.5\", hiddenSince = \\"1.6\")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R):
Float? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated(\\"Use maxByOrNull instead.\",
ReplaceWith(\\"this.maxByOrNull(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = \\"1.4\", errorSince =
\\"1.5\", hiddenSince = \\"1.6\")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) ->
R): Double? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated(\\"Use maxByOrNull instead.\",
ReplaceWith(\\"this.maxByOrNull(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince
= \\"1.4\", errorSince = \\"1.5\", hiddenSince = \\"1.6\")\npublic inline fun <R : Comparable<R>>
BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? {\n  return
maxByOrNull(selector)\n}\n\n@Deprecated(\\"Use maxByOrNull instead.\",
ReplaceWith(\\"this.maxByOrNull(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = \\"1.4\", errorSince =
\\"1.5\", hiddenSince = \\"1.6\")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R):
Char? {\n  return maxByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the largest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n *\n */\n@SinceKotlin(\\"1.4\")\npublic inline fun <T, R :
Comparable<R>> Array<out T>.maxByOrNull(selector: (T) -> R): T? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxVale =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxVale < v) {\n      maxElem
= e\n      maxVale = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the
largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n *\n */\n@SinceKotlin(\\"1.4\")\npublic inline fun <R :
Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) -> R): Byte? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxVale =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxVale < v) {\n      maxElem
= e\n      maxVale = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function
or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n *
*\n */\n@SinceKotlin(\\"1.4\")\npublic inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) ->
R): Short? {\n  if (isEmpty()) return null\n  var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if
(lastIndex == 0) return maxElem\n  var maxVale = selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e =
this[i]\n    val v = selector(e)\n    if (maxVale < v) {\n      maxElem = e\n      maxVale = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n *
*\n */\n@SinceKotlin(\\"1.4\")\npublic inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R):
Int? {\n  if (isEmpty()) return null\n  var maxElem = this[0]\n  val lastIndex
= this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxVale = selector(maxElem)\n  for (i in
1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxVale < v) {\n      maxElem = e\n      maxVale = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value
of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n *\n */\n@SinceKotlin(\\"1.4\")\npublic inline fun <R :
Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxVale =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxVale < v)

```

```

{\n      maxElem = e\n      maxValue = v\n    }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun  
<R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v)\n            {\n                maxElem = e\n                maxValue = v\n            }\n    }\n    return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v)\n            {\n                maxElem = e\n                maxValue = v\n            }\n    }\n    return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v)\n            {\n                maxElem = e\n                maxValue = v\n            }\n    }\n    return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue,

```

```

    v)\n } \n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`,
the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Double):
Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector:
(Float) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
* \n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector:
(T) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n
for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException\n * if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->
Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the\n * largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i

```



```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.maxOf(selector: (Long) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n
    maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline
fun <R : Comparable<R>> FloatArray.maxOf(selector: (Float) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOf(selector: (Double) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n
    maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.maxOf(selector: (Boolean) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n
    maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharArray.maxOf(selector: (Char) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOfOrNull(selector: (Byte) ->
Double): Double? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`

```

if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOrNull(selector: (Short) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOrNull(selector: (Int) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOrNull(selector: (Long) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector: (Float) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOrNull(selector: (Double) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector: (Boolean) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n
```

val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n } \n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector: (T) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOfOrNull(selector: (Long) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOfOrNull(selector: (Float) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOfOrNull(selector:
```

```

(Double) -> Float): Float? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Float): Float? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOfOrNull(selector: (Char) -> Float): Float? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxOfOrNull(selector: (T) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.maxOfOrNull(selector: (Byte) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.maxOfOrNull(selector: (Short) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> IntArray.maxOfOrNull(selector: (Int) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun <R : Comparable<R>> LongArray.maxOrNull(selector: (Long) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
FloatArray.maxOrNull(selector: (Float) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
DoubleArray.maxOrNull(selector: (Double) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
BooleanArray.maxOrNull(selector: (Boolean) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
CharArray.maxOrNull(selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
* @throws NoSuchElementException if the array is empty.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T, R> Array<out T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
* @throws NoSuchElementException if the array is empty.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.maxOfWith(comparator: Comparator<in R>, selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
* @throws

```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWith(comparator:\nComparator<in R>, selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n
```

```
    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided\n * [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWith(comparator:\nComparator<in R>, selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue\n    = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n
```

```
Returns the largest value according\n\nto the provided [comparator]\n * among all values produced by [selector] function applied to each element in the\narray.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWith(comparator:\nComparator<in R>, selector: (Long) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]\n * function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\n\ninline fun <R> FloatArray.maxOfWith(comparator: Comparator<in R>, selector: (Float) -> R): R {\n    if\n    (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all\n * values produced by [selector] function applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.maxOfWith(comparator:\nComparator<in R>, selector: (Double) -> R): R {\n    if (isEmpty())\n    throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return\n    maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.maxOfWith(comparator:\nComparator<in R>, selector: (Boolean) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n
```

```
    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among\n * all values produced by [selector] function applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n
```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R> Array<out T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n  if
(isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest
value according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n  if (isEmpty()) return
null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n  if
(isEmpty()) return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val

```



```

largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n  if\n  (isEmpty())\n  return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max,\n  e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the\n  provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n  LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator]\n  or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.maxWithOrNull(comparator:\n  Comparator<in Float>): Float? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max,\n  e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to\n  the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n  DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max =\n  e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided\n  [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n  BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max =\n  e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the\n  provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n  CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",\n  hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Double>.min(): Double? {\n  return\n  minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",\n  hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Float>.min(): Float? {\n  return\n  minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince\n  = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Array<out T>.min(): T?\n  {\n  return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",\n  hiddenSince = "1.6")\npublic fun ByteArray.min(): Byte? {\n  return minOrNull()\n}\n\n@Deprecated("Use\n  minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4",\n  errorSince = "1.5", hiddenSince = "1.6")\npublic fun ShortArray.min(): Short? {\n  return\n  minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",\n  hiddenSince = "1.6")\npublic fun IntArray.min(): Int? {\n  return minOrNull()\n}\n\n@Deprecated("Use\n  minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4",\n  errorSince = "1.5", hiddenSince\n  = "1.6")\npublic fun LongArray.min(): Long? {\n  return minOrNull()\n}\n\n@Deprecated("Use minOrNull\n  instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =\n  "1.5", hiddenSince = "1.6")\npublic fun FloatArray.min(): Float? {\n  return\n  minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\n  ReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",

```

```

hiddenSince = `1.6`)npublic fun DoubleArray.min(): Double? {n return
minOrNull()\n}\n\n@Deprecated(`Use minOrNull instead.`),
ReplaceWith(`this.minOrNull()\n`)n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince = `1.5`,
hiddenSince = `1.6`)npublic fun CharArray.min(): Char? {n return minOrNull()\n}\n\n@Deprecated(`Use
minByOrNull instead.`), ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince =
`1.4`, errorSince = `1.5`, hiddenSince = `1.6`)npublic inline fun <T,
R : Comparable<R>> Array<out T>.minBy(selector: (T) -> R): T? {n return
minByOrNull(selector)\n}\n\n@Deprecated(`Use minByOrNull instead.`),
ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince =
`1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>> ByteArray.minBy(selector: (Byte) -> R):
Byte? {n return minByOrNull(selector)\n}\n\n@Deprecated(`Use minByOrNull instead.`),
ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince =
`1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>> ShortArray.minBy(selector: (Short) -> R):
Short? {n return minByOrNull(selector)\n}\n\n@Deprecated(`Use minByOrNull instead.`),
ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince =
`1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>> IntArray.minBy(selector: (Int) -> R): Int?
{n return minByOrNull(selector)\n}\n\n@Deprecated(`Use
minByOrNull instead.`), ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince =
`1.4`, errorSince = `1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>>
LongArray.minBy(selector: (Long) -> R): Long? {n return minByOrNull(selector)\n}\n\n@Deprecated(`Use
minByOrNull instead.`), ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince =
`1.4`, errorSince = `1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>>
FloatArray.minBy(selector: (Float) -> R): Float? {n return minByOrNull(selector)\n}\n\n@Deprecated(`Use
minByOrNull instead.`), ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince =
`1.4`, errorSince = `1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>>
DoubleArray.minBy(selector: (Double) -> R): Double? {n return
minByOrNull(selector)\n}\n\n@Deprecated(`Use minByOrNull instead.`),
ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince
= `1.4`, errorSince = `1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>>
BooleanArray.minBy(selector: (Boolean) -> R): Boolean? {n return
minByOrNull(selector)\n}\n\n@Deprecated(`Use minByOrNull instead.`),
ReplaceWith(`this.minByOrNull(selector)\n`)n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince =
`1.5`, hiddenSince = `1.6`)npublic inline fun <R : Comparable<R>> CharArray.minBy(selector: (Char) -> R):
Char? {n return minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n * \n *\n */\n@SinceKotlin(`1.4`)npublic inline fun <T, R :
Comparable<R>> Array<out T>.minByOrNull(selector: (T) -> R): T? {n if (isEmpty()) return null\n var
minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex
== 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n
val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there
are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n * \n *\n */\n@SinceKotlin(`1.4`)npublic inline fun <R : Comparable<R>> ByteArray.minByOrNull(selector: (Byte) ->
R): Byte? {n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element

```

```

yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> ShortArray.minByOrNull(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> LongArray.minByOrNull(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> FloatArray.minByOrNull(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> DoubleArray.minByOrNull(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> BooleanArray.minByOrNull(selector: (Boolean) -> R): Boolean? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.minOf(selector: (Byte) -> Double): Double {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val
v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Double):
Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.minOf(selector: (Float) -> Double): Double {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) ->

```

```

Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun BooleanArray.minOf(selector: (Boolean) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun CharArray.minOf(selector: (Char) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun ByteArray.minOf(selector: (Byte) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun ShortArray.minOf(selector: (Short) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is NaN, the returned result is NaN. @throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun IntArray.minOf(selector: (Int) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

```

minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) -> Float): Float {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws\n    NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Float):\n    Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n * function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) ->\n    Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n    minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned\n * result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) ->\n    Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n    minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n * `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Float): Float {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws\n    NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out\n    T>.minOf(selector: (T) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n           \n            minValue = v\n        }\n    }\n    return minValue}\n\n/**\n * Returns the\n * smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>

```

```
ByteArray.minOf(selector: (Byte) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun <R : Comparable<R>> ShortArray.minOf(selector: (Short) -> R): R {\n  if (isEmpty()) throw\n  NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =\n  selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
IntArray.minOf(selector: (Int) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue =\n  selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]\n * function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is\n * empty.\n
```

```
empty.\n\n/**\n * Returns the smallest value among all values produced by [selector]\n * function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is\n * empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
LongArray.minOf(selector: (Long) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var\n  minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * @throws\n
```

```
NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun <R : Comparable<R>> FloatArray.minOf(selector: (Float) -> R): R {\n  if (isEmpty()) throw\n  NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =\n  selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the\n * array.\n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
DoubleArray.minOf(selector: (Double) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var\n  minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the\n * smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws\n
```

```
NoSuchElementException if the array is empty.\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * @throws\n
```

```
NoSuchElementException if the array is empty.\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * @throws\n
```

```
NoSuchElementException if the array is empty.\n
```



```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharArray.minOf(selector: (Char) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:
(T) -> Double): Double? {\n  if (isEmpty()) return
null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue
= minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) ->
Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element
in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -
> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.minOfOrNull(selector: (Int) -> Double): Double? {\n  if (isEmpty()) return null\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue =
minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -
> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the
array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) ->
Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`

```

if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
```

```
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of\n * values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector:
```

```
(Boolean) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) ->
```

```
Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns
```

```
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`\n * if there are no elements.\n
```

* \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:
```

```
(T) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex)\n    {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * \n
```

```
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array\n * or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n * is `NaN`, the returned\n * result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector:
```

```
(Byte) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -
```

```
> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the
```

```
smallest value among
```

```
all values produced by [selector] function\n * applied to each element in the array or `null` if there are no\n * elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) ->
```

```
Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the
```

smallest value among all values produced by [selector] function
* applied to each element in the array or `null` if there are no elements.
* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun LongArray.minOrNull(selector: (Long) -> Float): Float? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of\n * values produced by [selector] function is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOrNull(selector: (Float) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n
```

```
        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all\n * values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of\n * values produced by [selector] function is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOrNull(selector:\n(Double) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOrNull(selector:\n(Boolean) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n    1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOrNull(selector: (Char) ->\nFloat): Float? {\n    if (isEmpty())
```

```
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all\n * values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out\nT>.minOrNull(selector: (T) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for\n    (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
ByteArray.minOrNull(selector: (Byte) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced
```

```

by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ShortArray.minOfOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.minOfOrNull(selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> LongArray.minOfOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no
elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.minOfOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.minOfOrNull(selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> BooleanArray.minOfOrNull(selector: (Boolean) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no
elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.minOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array
is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.minOfWith(comparator:
Comparator<in
R>, selector: (Short) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (comparator.compare(minValue,
v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue
= selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (comparator.compare(minValue,
v)
> 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value according
to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.minOfWith(comparator: Comparator<in R>, selector: (Double) -> R): R {\n if
(isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n     minValue = v\n     }\n }\n
return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all
values produced by [selector] function applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n     minValue = v\n     }\n }\n
return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all
values produced by [selector] function applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if
(comparator.compare(minValue, v) > 0) {\n         minValue = v\n     }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among
all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if
(comparator.compare(minValue, v) > 0) {\n         minValue = v\n     }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if
(isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v =
selector(this[i])\n     if (comparator.compare(minValue, v) > 0) {\n         minValue = v\n     }\n }\n
return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n
if
(comparator.compare(minValue,
v) > 0) {\n         minValue = v\n     }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if

```

```
(comparator.compare(minValue, v) > 0) {\n    minValue = v\n } \n return minValue\n}\n\n/*\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each element in the array or `null` if there\n are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n    if (isEmpty()) return\n    null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        } \n    } \n    return minValue\n}\n\n/*\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
FloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float)\n-> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v\n        = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        } \n    } \n    return\n    minValue\n}\n\n/*\n * Returns the smallest value according to the provided [comparator]\n * among all values\n produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
DoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        } \n    } \n    return minValue\n}\n\n/*\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
BooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        } \n    } \n    return minValue\n}\n\n/*\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun <R> CharArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if\n    (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        } \n    } \n    return\n    minValue\n}\n\n/*\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is\n `NaN` returns `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun Array<out Double>.minOrNull(): Double? {\n    if\n    (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min,\n        e)\n    } \n    return min\n}\n\n/*\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any\n of elements is `NaN` returns `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun Array<out\n    Float>.minOrNull(): Float? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    } \n    return min\n}\n\n/*\n * Returns the smallest element or `null` if\n there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun Array<out T>.minOrNull():\n    T? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min\n        > e) min = e\n    } \n    return min\n}\n\n/*\n * Returns the smallest element or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun ByteArray.minOrNull(): Byte? {\n    if (isEmpty()) return null\n    var min =
```

```

this[0]\n for (i in 1..lastIndex) {\n     val e = this[i]\n     if (min > e) min = e\n } \n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\nShortArray.minOrNull(): Short? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e)\n            min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n    if (isEmpty()) return null\n    var min =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\nLongArray.minOrNull(): Long? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if\n * there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\nFloatArray.minOrNull(): Float? {\n    if\n    (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min =\n        minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull():\nDouble? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no\n * elements.\n */\n@SinceKotlin("1.4")\npublic fun CharArray.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return\n    min\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5",\n    hiddenSince = "1.6")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T? {\n    return\n    minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte? {\n    return\n    minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun ShortArray.minWith(comparator: Comparator<in Short>): Short? {\n    return\n    minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun IntArray.minWith(comparator: Comparator<in Int>):\nInt? {\n    return minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun LongArray.minWith(comparator: Comparator<in Long>): Long? {\n    return\n    minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun FloatArray.minWith(comparator: Comparator<in Float>): Float? {\n    return\n    minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun DoubleArray.minWith(comparator: Comparator<in Double>): Double?\n{\n    return minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull\n * instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4",\nerrorSince = "1.5", hiddenSince = "1.6")\npublic fun BooleanArray.minWith(comparator: Comparator<in\nBoolean>): Boolean? {\n    return minOrNull(comparator)\n}\n\n@Deprecated("Use minOrNull\n * instead.")\nReplaceWith("this.minOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4",\nerrorSince = "1.5", hiddenSince = "1.6")\npublic fun CharArray.minWith(comparator: Comparator<in Char>):\nChar? {\n    return minOrNull(comparator)\n}\n\n/**\n * Returns the first element having the smallest value

```



```

according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun
<T> Array<out T>.minWithOrNull(comparator: Comparator<in T>): T? {\n
if (isEmpty()) return null\n
var min
= this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the smallest value according to the provided [comparator]
or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun ByteArray.minWithOrNull(comparator:
Comparator<in Byte>): Byte? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first
element having the smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if
there are no elements.
*\n@SinceKotlin("1.4")\npublic fun IntArray.minWithOrNull(comparator:
Comparator<in Int>): Int? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val
e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first
element having the smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun FloatArray.minWithOrNull(comparator: Comparator<in Float>): Float? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.minWithOrNull(comparator: Comparator<in Double>):
Double? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>):
Boolean? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.
*\n@SinceKotlin("1.4")\npublic fun CharArray.minWithOrNull(comparator: Comparator<in Char>): Char? {\n
if (isEmpty()) return null\n
var min = this[0]\n
for (i in 1..lastIndex) {\n
val e = this[i]\n
if
(comparator.compare(min, e) > 0) min = e\n
}\n
return min\n}\n\n**\n * Returns `true` if the array has no
elements.
*\n * @sample samples.collections.Collections.Aggregates.none\n
*\npublic fun <T> Array<out
T>.none(): Boolean {\n
return isEmpty()\n}\n\n**\n * Returns `true` if the array has no elements.
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\npublic fun ByteArray.none(): Boolean {\n
return
isEmpty()\n}\n\n**\n * Returns `true` if the array has no elements.
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\npublic fun ShortArray.none(): Boolean {\n
return isEmpty()\n}\n\n**\n * Returns `true` if the array has no
elements.
*\n * @sample samples.collections.Collections.Aggregates.none\n
*\npublic fun IntArray.none():
Boolean {\n
return isEmpty()\n}\n\n**\n * Returns `true` if the array has no elements.
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\npublic fun LongArray.none(): Boolean {\n
return
isEmpty()\n}\n\n**\n * Returns `true` if the array has no elements.
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\npublic fun FloatArray.none(): Boolean {\n
return
isEmpty()\n}\n\n**\n * Returns `true` if the array has no elements.
*\n * @sample
samples.collections.Collections.Aggregates.none\n
*\npublic fun DoubleArray.none(): Boolean {\n
return

```

```

isEmpty()\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n\npublic fun BooleanArray.none():
Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n\npublic fun CharArray.none(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun <T> Array<out
T>.none(predicate: (T) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun ByteArray.none(predicate:
(Byte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun ShortArray.none(predicate: (Short) -> Boolean): Boolean {\n    for (element in this) if
(predicate(element)) return false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun
IntArray.none(predicate: (Int) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n
return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun LongArray.none(predicate:
(Long) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun FloatArray.none(predicate:
(Float)
-> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return true\n}\n\n/**\n *
Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun DoubleArray.none(predicate:
(Double) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun
BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {\n    for (element in this) if (predicate(element))
return false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n\npublic inline fun CharArray.none(predicate:
(Char) -> Boolean): Boolean {\n    for (element in this)
if (predicate(element)) return false\n    return true\n}\n\n/**\n * Performs the given [action] on each element and
returns the array itself afterwards.\n * \n * @SinceKotlin("1.4")\n * @kotlin.internal.InlineOnly\n * public inline fun <T>
Array<out T>.onEach(action: (T) -> Unit): Array<out T> {\n    return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n * \n *
@SinceKotlin("1.4")\n * @kotlin.internal.InlineOnly\n * public inline fun ByteArray.onEach(action: (Byte) ->
Unit): ByteArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n * \n * @SinceKotlin("1.4")\n * @kotlin.internal.InlineOnly\n * public inline fun ShortArray.onEach(action: (Short) ->
Unit): ShortArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns
the array itself afterwards.\n * \n * @SinceKotlin("1.4")\n * @kotlin.internal.InlineOnly\n * public inline fun
IntArray.onEach(action: (Int) -> Unit): IntArray {\n    return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n * \n *
@SinceKotlin("1.4")\n * @kotlin.internal.InlineOnly\n * public inline fun LongArray.onEach(action: (Long) ->
Unit): LongArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n

```

```

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEach(action: (Float) ->
Unit): FloatArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEach(action:
(Double) -> Unit): DoubleArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs
the given [action] on each element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEach(action: (Boolean)
-> Unit): BooleanArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEach(action: (Char) ->
Unit): CharArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n *
@param [action] function that takes the index of an element and the element itself\n * and performs the action on
the element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Array<out T>.onEachIndexed(action: (index: Int, T) -> Unit): Array<out T> {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.onEachIndexed(action:
(index: Int, Byte) -> Unit): ByteArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the
given [action] on each element, providing sequential index with the element,\n * and returns the array itself
afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.onEachIndexed(action:
(index: Int, Short) -> Unit): ShortArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the
given [action] on each element, providing sequential index with the element,\n * and returns the array itself
afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.onEachIndexed(action: (index: Int, Int) -> Unit): IntArray {\n    return apply { forEachIndexed(action)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element
itself\n * and performs the action on the element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.onEachIndexed(action: (index: Int, Long) -> Unit):
LongArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each
element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param
[action] function that takes the index of an element and the element itself\n * and performs the action on the
element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.onEachIndexed(action: (index: Int, Float) -> Unit): FloatArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEachIndexed(action:
(index: Int, Double) -> Unit): DoubleArray {\n    return apply { forEachIndexed(action)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element
itself\n * and performs the action on the element.\n *
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.onEachIndexed(action: (index: Int, Boolean) -> Unit): BooleanArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index

```

with the element, `\n *` and returns the array itself afterwards. `\n * @param [action]` function that takes the index of an element and the element itself `\n *` and performs the action on the element. `\n`

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEachIndexed(action:  
(index: Int, Char) -> Unit): CharArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates  
value starting with the first element and applying [operation] from left to right\n * to current accumulator value and  
each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function  
that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n *  
@sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun <S, T : S> Array<out  
T>.reduce(operation: (acc: S, T) -> S): S {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty  
array can't be reduced.")\n    var accumulator: S = this[0]\n    for (index in 1..lastIndex) {\n        accumulator =  
operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the  
first element and applying [operation] from left to right\n * to current accumulator  
value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an  
expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param  
[operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator  
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun  
ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in  
in
```

```
an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *  
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next  
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun  
ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
```

```
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next  
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun  
IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
```

```
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next  
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun  
LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
```

@param [operation] function that takes current accumulator value and an element,`\n *` and calculates the next accumulator value.`\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n`

* Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 * \npublic inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {
 * if (isEmpty())
 * throw UnsupportedOperationException("Empty array can't be reduced.")
 * var accumulator = this[0]
 * for (index in 1..lastIndex) {
 * accumulator = operation(index, accumulator, this[index])
 * }
 * return accumulator
 * }
 * \n\n/**

* Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 * \npublic inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {
 * if (isEmpty())
 * throw UnsupportedOperationException("Empty array can't be reduced.")
 * var accumulator = this[0]
 * for (index in 1..lastIndex) {
 * accumulator = operation(index, accumulator, this[index])
 * }
 * return accumulator
 * }
 * \n\n/**

* Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 * \npublic inline fun LongArray.reduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): Long {
 * if (isEmpty())
 * throw UnsupportedOperationException("Empty array can't be reduced.")
 * var accumulator = this[0]
 * for (index in 1..lastIndex) {
 * accumulator = operation(index, accumulator, this[index])
 * }
 * return accumulator
 * }
 * \n\n/**

* Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 * \npublic inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {
 * if (isEmpty())
 * throw UnsupportedOperationException("Empty array can't be reduced.")
 * var accumulator = this[0]
 * for (index in 1..lastIndex) {
 * accumulator = operation(index, accumulator, this[index])
 * }
 * return accumulator
 * }
 * \n\n/**

* Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 * \npublic inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {
 * if (isEmpty())
 * throw UnsupportedOperationException("Empty array can't be reduced.")
 * var accumulator = this[0]
 * for (index in 1..lastIndex) {
 * accumulator = operation(index, accumulator, this[index])
 * }
 * return accumulator
 * }
 * \n\n/**

accumulator\n}\n\n/**\n

* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {\n    if (isEmpty())\n        throw
```

```
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return
```

accumulator\n}\n\n/**\n

* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n    if (isEmpty())\n        throw
```

```
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return
```

accumulator\n}\n\n/**\n

* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.reduceOrNull\n *\n@SinceKotlin("1.4")\npublic inline fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n    if (isEmpty())\n        return
```

```
null\n    var accumulator: S = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n
```

* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the

original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.reduceOrNull\n *\n@SinceKotlin("1.4")\npublic inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in
```

```
1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return
```

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element

itself,\n * and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.reduceOrNull\n *\n@SinceKotlin("1.4")\npublic inline fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index,
```

```
accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n
```

* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the

```

index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic
inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,
accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element with its index
in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun LongArray.reduceIndexedOrNull(operation: (index: Int, acc: Long,
Long) -> Long): Long? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator
value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,
accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
the first element and applying [operation] from left to right\n * to current accumulator value and each element with
its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc:
Double, Double) -> Double): Double? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns
`null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {\n if
(isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element
with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,
accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n

```



```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Array<out T>.reduceOrNull(operation:
(acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in
1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation]
function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n
return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from
left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {\n if (isEmpty())\n return null\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array
is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is
empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
inline fun LongArray.reduceOrNull(operation: (acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return
null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator,
this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null`
if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var
accumulator
= this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return
accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return

```

```

null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(accumulator,
this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and
applying
[operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array
is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {\n  if (isEmpty())\n  return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator =
operation(accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n  if (isEmpty())\n  return null\n  var
accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(accumulator, this[index])\n
  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\npublic inline fun <S, T : S> Array<out T>.reduceRight(operation: (T, acc: S) -> S): S {\n  var index =
lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var
accumulator: S = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n
  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\npublic inline fun
ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {\n  var index = lastIndex\n  if (index < 0)
throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\npublic inline fun ShortArray.reduceRight(operation:
(Short, acc: Short) -> Short): Short {\n  var
index = lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n
var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n
  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator

```

value,`\n *` and calculates the next accumulator value.`\n * \n * @sample`

`samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {``\n var index = lastIndex``\n if (index < 0) throw``UnsupportedOperationException("Empty array``can't be reduced.")``\n var accumulator = get(index--)``\n while (index >= 0) {``\n accumulator =``operation(get(index--), accumulator)``\n }``\n return accumulator``\n }``\n }``\n /**``\n * Accumulates value starting with the``last element and applying [operation] from right to left``\n * to each element and current accumulator value.``\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,``\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.``\n * \n * @param [operation] function that``takes an element and current accumulator value,``\n * and calculates the next accumulator value.``\n * \n * @sample``samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {``\n var index = lastIndex``\n if (index < 0) throw``UnsupportedOperationException("Empty array can't be reduced.")``\n var accumulator = get(index--)``\n while (index >=``0) {``\n accumulator = operation(get(index--), accumulator)``\n }``\n return accumulator``\n }``\n }``\n /**``\n * Accumulates value starting with the last element and applying [operation] from right to left``\n * to each element and current accumulator value.``\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,``\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.``\n * \n * @param [operation] function that takes an element and current accumulator value,``\n * and calculates the next accumulator value.``\n * \n * @sample``samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {``\n var index = lastIndex``\n if (index < 0) throw``UnsupportedOperationException("Empty array can't be reduced.")``\n var accumulator = get(index--)``\n while (index >= 0) {``\n accumulator = operation(get(index--), accumulator)``\n }``\n return accumulator``\n }``\n }``\n /**``\n * Accumulates value starting with the last element and applying [operation] from``right to left``\n * to each element and current accumulator value.``\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,``\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.``\n * \n * @param [operation] function that takes an element and current accumulator value,``\n * and calculates the next accumulator value.``\n * \n * @sample``samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {``\n var index = lastIndex``\n if (index < 0) throw``UnsupportedOperationException("Empty array can't be reduced.")``\n var accumulator = get(index--)``\n while (index >= 0) {``\n accumulator = operation(get(index--), accumulator)``\n }``\n return accumulator``\n }``\n }``\n /**``\n * Accumulates value starting with the``last element and applying [operation] from right to left``\n * to each element and current accumulator value.``\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,``\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.``\n * \n * @param [operation] function that takes an element and current accumulator value,``\n * and calculates the next accumulator value.``\n * \n * @sample``samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {``\n var index = lastIndex``\n if (index < 0) throw``UnsupportedOperationException("Empty array can't be reduced.")``\n var accumulator =``get(index--)``\n while (index >= 0) {``\n accumulator = operation(get(index--), accumulator)``\n }``\n return``accumulator``\n }``\n }``\n /**``\n * Accumulates value starting with the last element and applying [operation] from right to``left``\n * to each``element and current accumulator value.``\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,``\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.``\n * \n * @param [operation] function that takes an element and current accumulator value,``\n * and calculates the next accumulator value.``\n * \n * @sample``samples.collections.Collections.Aggregates.reduceRight``\n * \n public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {``\n var index = lastIndex``\n if (index < 0)`

```

throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use\n * [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]\n * function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the\n * next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic\ninline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n    var index =\n    lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var\n    accumulator: S = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),\n        accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last\n * element and applying [operation] from right to left\n * to each element with its index in the original\n * array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty\n * in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is\n * empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current\n * accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun\nByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]\n * from right to left\n * to each\n * element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array\n * is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It\n * returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element,\n * the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun\nShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {\n    var index =\n    lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var\n    accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),\n        accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last\n * element and applying\n * [operation] from right to left\n * to each element with its index in the original array and current accumulator\n * value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please\n * use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]\n * function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the\n * next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic\ninline fun IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with\n * the last element and applying [operation] from right to left\n * to each element with its index in the original array\n * and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator\n * value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun

```

```

LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.
* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.
* @sample samples.collections.Collections.Aggregates.reduceRight
public inline fun

FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

* Accumulates value starting with the last element and applying [operation]
* from right to left
* to each element with its index in the original array and current accumulator value.
* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.
* @sample samples.collections.Collections.Aggregates.reduceRight
public inline fun

DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.
* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.
* @sample samples.collections.Collections.Aggregates.reduceRight
public inline fun

BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.
* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.
* @sample samples.collections.Collections.Aggregates.reduceRight
public inline fun

CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.
* Returns `null` if the array is empty.
* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.
* @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```

```

*^@SinceKotlin("1.4")\npublic inline fun <S, T : S> Array<out T>.reduceRightIndexedOrNull(operation:
(index: Int, T, acc: S) -> S): S? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator: S =
get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n
--index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying
[operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n
*\n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element,
the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic inline fun
ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right
to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null`
if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic inline fun
ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the
array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic inline fun
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int? {\n  var index = lastIndex\n
if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with the last element and applying [operation] from right to left\n * to each element with its index in
the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes the index of an element, the element itself and current accumulator value,\n
*\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic inline fun
LongArray.reduceRightIndexedOrNull(operation: (index: Int, Long, acc: Long) -> Long): Long? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
*\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic
inline fun FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
*\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^@SinceKotlin("1.4")\npublic inline fun

```

```

DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double):
Double? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * @SinceKotlin("1.4")\npublic inline fun
BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {\n
var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index
>= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * @SinceKotlin("1.4")\npublic inline fun
CharArray.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element
and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
* \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Array<out T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n  var index = lastIndex\n  if (index < 0)
return null\n  var accumulator: S = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--),
accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
* \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {\n  var index = lastIndex\n  if (index
< 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
* \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
* \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun

```

```

IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element
and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the
array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and
calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceRightOrNull(operation: (Char,
acc: Char) -> Char): Char? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator =
get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n
* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an

```



```

element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n *^\n@SinceKotlin("1.4")\npublic inline fun <T, R>
Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n  if (isEmpty()) return
listOf(initial)\n  val result
= ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n *
to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*^\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFold(initial: R,
operation: (acc: R, Byte) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result
= ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n *
to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*^\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFold(initial: R,
operation: (acc: R, Short) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val
result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n *
to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*^\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFold(initial: R,
operation: (acc: R, Int) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n
val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n *
to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*^\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFold(initial: R,
operation: (acc: R, Long) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n
val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n *
to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*^\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.runningFold(initial: R,

```

```

operation: (acc: R, Float) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n
  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
  accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n* to
each element and current accumulator value that starts with [initial] value.\n* \n* Note that `acc` value passed to
[operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n*
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.runningFold(initial: R,
operation: (acc: R, Double) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n
  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n
  accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n  return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n* to
each element and current accumulator value that starts with [initial] value.\n* \n* Note that `acc` value passed to
[operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n*
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.runningFold(initial:
R, operation: (acc: R, Boolean) -> R): List<R> {\n  if (isEmpty())
  return listOf(initial)\n  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n
for (element in this) {\n  accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n
return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n* to each element and current accumulator value that starts with [initial] value.\n* \n*
Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the
previous value in resulting list.\n* \n* @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n* \n* @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.runningFold(initial: R,
operation: (acc: R, Char) -> R): List<R> {\n  if (isEmpty())
  return listOf(initial)\n  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n
for (element in this) {\n  accumulator = operation(accumulator, element)\n  result.add(accumulator)\n  }\n
return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n* to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n* \n* Note that `acc` value passed to [operation] function should not be mutated;\n*
otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the
index of an element, current accumulator value\n* and the element itself, and calculates the next accumulator
value.\n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <T, R> Array<out T>.runningFoldIndexed(initial: R,
operation: (index: Int, acc: R, T) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result =
ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for (index in indices) {\n
  accumulator = operation(index, accumulator, this[index])\n  result.add(accumulator)\n  }\n  return
result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n* to each element, its index in the original array and current accumulator value that starts with [initial]
value.\n* \n* Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would
affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element,
current accumulator value\n* and the element itself, and calculates the next accumulator value.\n* \n* @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun <R> ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <R> IntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <R> LongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

values generated by applying [operation] from left to right\n *
to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n *
Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

DoubleArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns
a list containing successive accumulation values generated by applying [operation] from left to right\n * to each
element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous
value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator
value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

BooleanArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for (index in indices) {\n accumulator = operation(index,
accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element,
its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc`
value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n *
and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

CharArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial)
}\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator,
this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and the element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
```

Array<out T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n if (isEmpty()) return emptyList()\n var
accumulator: S = this[0]\n val result = ArrayList<S>(size).apply { add(accumulator) }\n for (index in 1 until
size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return
result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduce(operation:
```

(acc: Byte, Byte) -> Byte): List<Byte> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val

```

result = ArrayList<Byte>(size).apply { add(accumulator) } for (index in 1 until size) {
    accumulator = operation(accumulator, this[index])
    result.add(accumulator)
} return
result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ShortArray.runningReduce(operation:
(acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator =
operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduce(operation: (acc:
Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result =
ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator =
operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun LongArray.runningReduce(operation:
(acc: Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator =
operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.runningReduce(operation:
(acc: Float, Float) -> Float): List<Float> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Float>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator =
operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.runningReduce(operation:
(acc: Double, Double) -> Double): List<Double> {
    if (isEmpty()) return emptyList()
    var accumulator =
this[0]
    val result = ArrayList<Double>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator =
operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

*
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline

```

```

fun BooleanArray.runningReduce(operation: (acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Boolean>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.runningReduce

*/

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun CharArray.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.runningReduce

*/

@SinceKotlin("1.4")
public inline fun <S, T : S> Array<out T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (isEmpty()) return emptyList()
    var accumulator: S = this[0]
    val result = ArrayList<S>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.runningReduce

*/

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Byte>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.runningReduce

*/

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.runningReduce

*/

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return

```

emptyList()\n

```
var accumulator = this[0]\n val result = ArrayList<Int>(size).apply { add(accumulator) }\n for (index in 1\n until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying\n [operation] from left to right\n * to each element, its index in the original array and current accumulator value that\n starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element,\n current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample\n samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
LongArray.runningReduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): List<Long> {\n if\n (isEmpty()) return emptyList()\n var accumulator =
```

```
this[0]\n val result = ArrayList<Long>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return\n result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from\n left to right\n * to each element, its index in the original array and current accumulator value that starts with the first\n element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator\n value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {\n if\n (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result =
```

```
ArrayList<Float>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =\n operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this\n array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {\n if\n (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Double>(size).apply
```

```
{ add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator,\n this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive\n accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original\n array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation]\n function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the\n next accumulator value.\n * \n * @sample\n samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean>\n {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result =
```

```
ArrayList<Boolean>(size).apply {\n add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator,\n this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive\n accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original\n array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation]\n function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the\n next accumulator value.\n * \n * @sample\n samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```

CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty())
    return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply { add(accumulator) }
    for (index
    in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an
 element, and calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R>
Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {
    return runningFold(initial,
    operation)
}

/**
 * Returns a list containing successive accumulation values generated
 by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial]
 value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value
 and an element, and calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R>
ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {
    return runningFold(initial,
    operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation]
 from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should
 not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and
 calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R>
ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {
    return
    runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by
 applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial]
 value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value
 and an element, and calculates the next
 accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R>
IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {
    return runningFold(initial,
    operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation]
 from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in
 resulting list.
 * @param [operation] function that takes current accumulator value and an element, and
 calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline
fun <R>
LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {
    return runningFold(initial,
    operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation]
 from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in
 resulting list.
 * @param [operation] function that takes current accumulator value and an element, and

```


calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {
    return runningFold(initial, operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {
    return runningFold(initial, operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {
    return runningFold(initial, operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@kotlin.internal.InlineOnly
public inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {
    return runningFold(initial, operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R> Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.scanIndexed(initial:
R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    return runningFoldIndexed(initial,
operation)
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes the index of an

element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n
* Returns a list containing successive accumulation values generated by applying [operation] from left to rightn
* to each element, its index in the original array and current accumulator value that starts with [initial] value.n
* Note that `acc` value passed to [operation] function should not be mutated;n
* otherwise it would affect the previous value in resulting list.n
* @param [operation] function that takes the index of an element, current accumulator value
* and the element itself, and calculates the next accumulator value.n
* @sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> IntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n
* Returns a list containing successive accumulation values generated by applying [operation] from left to rightn
* to each element, its index in the original array and current accumulator value that starts with [initial] value.n
* Note that `acc` value passed to [operation] function should not be mutated;n
* otherwise it would affect the previous value in resulting list.n
* @param [operation] function that takes the index of an element, current accumulator value
* and the element itself, and calculates the next accumulator value.n
* @sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> LongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n
* Returns a list containing successive accumulation values generated by applying [operation] from left to rightn
* to each element, its index in the original array and current accumulator value that starts with [initial] value.n
* Note that `acc` value passed to [operation] function should not be mutated;n
* otherwise it would affect the previous value in resulting list.n
* @param [operation] function that takes the index of an element, current accumulator value
* and the element itself, and calculates the next accumulator value.n
* @sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n
* Returns a list containing successive accumulation values generated by applying [operation] from left to rightn
* to each element, its index in the original array and current accumulator value that starts with [initial] value.n
* Note that `acc` value passed to [operation] function should not be mutated;n
* otherwise it would affect the previous value in resulting list.n
* @param [operation] function that takes the index of an element, current accumulator value
* and the element itself, and calculates the next accumulator value.n
* @sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n
* Returns a list containing successive accumulation values generated by applying [operation] from left to rightn
* to each element, its index in the original array and current accumulator value that starts with [initial] value.n
* Note that `acc` value passed to [operation] function should not be mutated;n
* otherwise it would affect the previous value in resulting list.n
* @param [operation] function that takes the index of an element, current accumulator value
* and the element itself, and calculates the next accumulator value.n
* @sample samples.collections.Collections.Aggregates.scan
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {n

```

```

return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n
* and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n    return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Array<out T>.sumBy(selector: (T) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince
= "1.5")\npublic inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {\n    var sum: Int = 0\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ShortArray.sumBy(selector: (Short) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
IntArray.sumBy(selector: (Int) -> Int): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
LongArray.sumBy(selector: (Long) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
FloatArray.sumBy(selector: (Float) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied
to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
DoubleArray.sumBy(selector: (Double) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
BooleanArray.sumBy(selector: (Boolean) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
inline fun CharArray.sumBy(selector: (Char) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",

```



```

array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sumOf(selector: (Short) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Double):
Double {\n    var sum: Double = 0.toDouble()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.sumOf(selector: (Long) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float)
-> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sumOf(selector: (Double) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector:
(Boolean) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.sumOf(selector: (Char) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Array<out T>.sumOf(selector: (T) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.sumOf(selector: (Byte) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function

```

applied to each element in the array.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nShortArray.sumOf(selector:
```

```
(Short) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nIntArray.sumOf(selector: (Int) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=\n        selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nLongArray.sumOf(selector: (Long) -> Int):
```

```
Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nFloatArray.sumOf(selector: (Float) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=\n        selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nDoubleArray.sumOf(selector: (Double) -> Int): Int {\n
```

```
    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nBooleanArray.sumOf(selector: (Boolean) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nCharArray.sumOf(selector: (Char) -> Int): Int {\n    var sum: Int
```

```
= 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\n<T> Array<out T>.sumOf(selector: (T) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)\n    {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nByteArray.sumOf(selector: (Byte) -> Long): Long {\n    var sum: Long = 0.toLong()\n
```

```
    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nShortArray.sumOf(selector: (Short) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nIntArray.sumOf(selector: (Int) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all\n values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nLongArray.sumOf(selector: (Long) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nFloatArray.sumOf(selector: (Float) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element\n in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced\n by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nDoubleArray.sumOf(selector: (Double) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)\n {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by\n [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nBooleanArray.sumOf(selector: (Boolean) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element\n in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced\n by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nCharArray.sumOf(selector: (Char) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt\n {\n    var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {\n    var\n sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
```

```

s::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {\n  var sum: UInt = 0.toUInt()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {\n  var
sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n *
Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {\n  var sum: UInt = 0.toUInt()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {\n  var
sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n *
Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt {\n  var sum: UInt = 0.toUInt()\n  for (element
in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt
{\n  var sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return
sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfUInt")\n@\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {\n  var
sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n *
Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfULong")\n@\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong):
ULong {\n  var sum: ULong = 0.toULong()\n  for (element in this)
{\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@\n@SinceKotlin("1.5")\n@\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@\n@OverloadResolution
ByLambdaReturnType\n@\n@kotlin.jvm.JvmName("sumOfULong")\n@\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong
{\n  var sum: ULong = 0.toULong()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the

```



```

array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector:
(Short) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {\n
var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.sumOf(selector: (Double) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n\n*\npublic fun <T : Any>
Array<T?>.requireNonNulls(): Array<T> {\n    for (element in this) {\n        if (element == null) {\n            throw
IllegalArgumentException("null element found in $this.")\n        }\n    }\n    @Suppress("UNCHECKED_CAST")\n    return this as Array<T>\n}\n\n/**\n * Splits the original array into pair
of lists,\n\n*\n

```

where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

```

samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun <T> Array<out T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val second = ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded true`,\n * while *second* list contains elements for which [predicate] yielded false`.\n * \n * @sample

```

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n val first = ArrayList<Byte>()\n val second = ArrayList<Byte>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n val first = ArrayList<Short>()\n val second = ArrayList<Short>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n val first = ArrayList<Int>()\n val second = ArrayList<Int>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n val first = ArrayList<Long>()\n val second = ArrayList<Long>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n val first = ArrayList<Float>()\n val second = ArrayList<Float>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n val first = ArrayList<Double>()\n val second = ArrayList<Double>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where **first** list contains elements for which [predicate] yielded `true``,\n * while **second** list contains elements for which [predicate] yielded `false``.\n * \n * @sample

`@sample` samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n

```

*public inline fun BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>,
List<Boolean>> {\n  val first = ArrayList<Boolean>()\n  val second = ArrayList<Boolean>()\n  for (element in
this) {\n    if (predicate(element)) {\n      first.add(element)\n    } else {\n      second.add(element)\n
}\n  }\n  return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list
contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which
[predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n *public inline fun
CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {\n  val first =
ArrayList<Char>()\n  val second = ArrayList<Char>()\n  for (element in this) {\n    if (predicate(element)) {\n
first.add(element)\n
    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first, second)\n}\n\n/**\n * Returns a list of
pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has
length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *public
infix fun <T, R> Array<out T>.zip(other: Array<out R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 -> t1 to
t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R> ByteArray.zip(other: Array<out R>):
List<Pair<Byte, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` array and the [other] array with the same index.\n * The returned
list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *public
infix fun <R> ShortArray.zip(other: Array<out R>): List<Pair<Short, R>> {\n  return zip(other) { t1, t2
-> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the
same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R> IntArray.zip(other: Array<out R>):
List<Pair<Int, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R>
LongArray.zip(other: Array<out R>): List<Pair<Long, R>> {\n  return zip(other) { t1, t2 ->
t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the
same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R> FloatArray.zip(other: Array<out
R>): List<Pair<Float, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R>
DoubleArray.zip(other: Array<out R>): List<Pair<Double, R>> {\n  return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R> BooleanArray.zip(other: Array<out R>):
List<Pair<Boolean, R>> {\n  return zip(other) {
t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array
with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *public infix fun <R> CharArray.zip(other: Array<out R>):
List<Pair<Char, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the
elements of `this` array and the [other] array with the same index\n * using the provided [transform] function
applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *public inline fun <T, R, V> Array<out
T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V):

```

```

List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements
 * of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each
 * pair of elements.
 * The returned list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array
 * with the same index
 * using the provided [transform] function applied to each
 * pair of elements.
 * The returned list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array
 * with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned
 * list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until
size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built
 * from the elements of `this` array and the [other] array with the same index
 * using the provided [transform]
 * function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array
 * with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned
 * list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements
 * of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each
 * pair of elements.
 * The returned list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns
 * a list of values built from the elements of `this` array and the [other] array with the same index
 * using the
 * provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest
 * collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline
fun <R, V>
BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the
 * [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned
 * list has length of the shortest collection.
 *
 * @sample
 * samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public inline fun <R, V>
CharArray.zip(other:

```

```

Array<out R>, transform: (a: Char, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list =
ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return
list}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Array<out T>.zip(other:
Iterable<R>): List<Pair<T, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R>
ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {\n    return zip(other) { t1,
t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ShortArray.zip(other: Iterable<R>):
List<Pair<Short, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R>
IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> LongArray.zip(other: Iterable<R>): List<Pair<Long, R>> {\n    return zip(other) { t1, t2 -
> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the
same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> FloatArray.zip(other: Iterable<R>):
List<Pair<Float, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R>
DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {\n    return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements
of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R>
BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {\n    return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> CharArray.zip(other: Iterable<R>):
List<Pair<Char, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the
elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function
applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n
val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i =
0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with
the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list
has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V>
ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n

```

```

    var i = 0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other]
collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n
if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline fun <R, V> IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {\n    val
arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n
for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the
same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has
length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
LongArray.zip(other: Iterable<R>,
transform: (a: Long, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n
if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline
fun <R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {\n    val arraySize =
size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element
in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other]
collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {\n    val arraySize = size\n    val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other)
{\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline
fun <R, V> BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {\n    val arraySize =
size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element
in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the
same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has
length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10),

```

```

arraySize))\n  var i = 0\n  for (element in other) {\n    if (i >= arraySize) break\n    list.add(transform(this[i++], element))\n  }\n  return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun ByteArray.zip(other: ByteArray): List<Pair<Byte, Byte>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun IntArray.zip(other: IntArray): List<Pair<Int, Int>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun LongArray.zip(other: LongArray): List<Pair<Long, Long>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun FloatArray.zip(other: FloatArray): List<Pair<Float, Float>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun DoubleArray.zip(other: DoubleArray): List<Pair<Double, Double>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun BooleanArray.zip(other: BooleanArray): List<Pair<Boolean, Boolean>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun CharArray.zip(other: CharArray): List<Pair<Char, Char>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {\n  val size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {\n  val size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {\n  val size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided

```

```

[transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n *
@sample samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun <V>
LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n *
@sample samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun <V>
FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun <V>
DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using
the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun
<V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {\n    val size =
minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the
[other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n *
The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun <V>
CharArray.zip(other:
CharArray, transform: (a: Char, b: Char) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list =
ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return
list\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix]
and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in
which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to
`"...`").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n *^\npublic fun <T, A :
Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = ``, ``, prefix: CharSequence = ``, postfix:
CharSequence = ``, limit: Int = -1, truncated: CharSequence = `"...`", transform: ((T) -> CharSequence)? = null): A {\n
    buffer.append(prefix)\n    var count = 0\n    for
(element in this) {\n        if (++count > 1) buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n
            buffer.appendElement(element, transform)\n        } else break\n    }\n    if (limit >= 0 && count > limit)
buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer\n}\n\n/**\n * Appends the string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will
be appended, followed by the [truncated] string (which defaults to `"...`").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n *^\npublic fun <A : Appendable>
ByteArray.joinTo(buffer: A, separator: CharSequence = ``, ``, prefix: CharSequence = ``, postfix: CharSequence =
``, limit: Int = -1, truncated: CharSequence = `"...`", transform: ((Byte) -> CharSequence)? = null): A {\n
    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            if (transform != null)\n                buffer.append(transform(element))\n            else\n                buffer.append(element.toString())\n        } else break\n    }

```



```

}\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer}\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A :
Appendable> ShortArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix:
CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...", transform:
((Short) -> CharSequence)? = null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n
if (++count > 1) buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else
break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer}\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A : Appendable> IntArray.joinTo(buffer:
A, separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1,
truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): A {\n  buffer.append(prefix)\n  var
count = 0\n  for (element in this) {\n    if (++count > 1) buffer.append(separator)\n    if (limit < 0 || count <=
limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit)
buffer.append(truncated)\n  buffer.append(postfix)\n  return buffer}\n}\n\n/**\n * Appends the string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit],
in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults
to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A :
Appendable> LongArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix:
CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? =
null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer}\n}\n\n/**\n * Appends the string from all the elements separated using
[separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can
specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed
by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A : Appendable>
FloatArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence =
\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Float) -> CharSequence)? = null): A {\n
buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1) buffer.append(separator)\n
    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >=
0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return buffer}\n}\n\n/**\n * Appends
the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n *
\n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first
[limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A : Appendable>
DoubleArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix:

```

```

CharSequence = "\\\"", limit: Int = -1, truncated: CharSequence = "...\"", transform: ((Double) -> CharSequence)? =
null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else
break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer}\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> BooleanArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\\\"",
postfix: CharSequence = "\\\"", limit: Int = -1, truncated: CharSequence = "...\"", transform: ((Boolean) ->
CharSequence)? = null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1) buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit)
buffer.append(truncated)\n  buffer.append(postfix)\n  return buffer}\n}\n\n/**\n * Appends the string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will
be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A : Appendable>
CharArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\\\"", postfix: CharSequence =
"\",
limit: Int = -1, truncated: CharSequence = "...\"", transform: ((Char) -> CharSequence)? = null): A {\n
buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1) buffer.append(separator)\n
    if (limit < 0 || count <= limit) {\n      if (transform != null)\n        buffer.append(transform(element))\n
    else\n      buffer.append(element)\n    } else break\n  }\n  if (limit >= 0 && count > limit)
buffer.append(truncated)\n  buffer.append(postfix)\n  return buffer}\n}\n\n/**\n * Creates a string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will
be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n * \npublic
fun <T> Array<out T>.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\\\"", postfix:
CharSequence = "\\\"", limit: Int = -1, truncated: CharSequence = "...\"", transform: ((T) -> CharSequence)? = null):
String {\n  return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,
transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n * \npublic
fun ByteArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\\\"", postfix:
CharSequence = "\\\"", limit: Int = -1, truncated: CharSequence = "...\"", transform: ((Byte) -> CharSequence)?
= null): String {\n  return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,
transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n * \npublic
fun ShortArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\\\"", postfix:
CharSequence = "\\\"", limit: Int = -1, truncated: CharSequence = "...\"", transform: ((Short) -> CharSequence)? =
null): String {\n  return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,

```

```

transform).toString()\n\n/**\n * Creates a string from all the elements separated using [separator] and using
the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative
value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun IntArray.joinToString(separator: CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix:
CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Int) -> CharSequence)? = null):
String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,
transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the
[truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun LongArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated:
CharSequence = \"...\", transform: ((Long) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun FloatArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence
= \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Float) ->
CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,
transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun DoubleArray.joinToString(separator: CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix:
CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Double) -> CharSequence)? =
null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix,
limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using
[separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can
specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed
by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun
BooleanArray.joinToString(separator: CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence =
\"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Boolean) -> CharSequence)? = null): String {\n
return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates
a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n
* If the collection could
be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n
*/\npublic fun CharArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated:
CharSequence = \"...\", transform: ((Char) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates an [Iterable] instance that
wraps the original array returning its elements when being iterated.\n */\npublic fun <T> Array<out T>.asIterable():
Iterable<T> {\n    if (isEmpty()) return emptyList()\n    return Iterable { this.iterator() }\n}\n\n/**\n * Creates an
[Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic

```

```

fun ByteArray.asIterable(): Iterable<Byte> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun ShortArray.asIterable(): Iterable<Short> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun IntArray.asIterable(): Iterable<Int> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun LongArray.asIterable(): Iterable<Long> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun FloatArray.asIterable(): Iterable<Float> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun DoubleArray.asIterable(): Iterable<Double> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun BooleanArray.asIterable(): Iterable<Boolean> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.

fun CharArray.asIterable(): Iterable<Char> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample
samples.collections.Sequences.Building.sequenceFromArray

fun <T> Array<out T>.asSequence(): Sequence<T> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun ByteArray.asSequence(): Sequence<Byte> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun ShortArray.asSequence(): Sequence<Short> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample
samples.collections.Sequences.Building.sequenceFromArray

fun IntArray.asSequence(): Sequence<Int> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun LongArray.asSequence(): Sequence<Long> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun FloatArray.asSequence(): Sequence<Float> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample
samples.collections.Sequences.Building.sequenceFromArray

fun DoubleArray.asSequence(): Sequence<Double> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun BooleanArray.asSequence(): Sequence<Boolean> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

// @sample samples.collections.Sequences.Building.sequenceFromArray

fun CharArray.asSequence(): Sequence<Char> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}
// Returns an average value

```

```

of elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun Array<out Byte>.average():
Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of
elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Array<out Short>.average():
Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of
elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Array<out Int>.average(): Double
{\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun
Array<out Long>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun
Array<out Float>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun
Array<out Double>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this)
{\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN
else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\npublic fun
ByteArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns
an average value of elements in the array.\n */\npublic fun ShortArray.average(): Double {\n    var sum: Double =
0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\npublic
fun IntArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns
an average value of elements in the array.\n */\npublic fun LongArray.average():
Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns an average value of
elements in the array.\n */\npublic fun FloatArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int
= 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN
else sum / count\n}\n\n/* Returns an average value of elements in the array.\n */\npublic fun
DoubleArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/* Returns
the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Array<out
Byte>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Array<out Short>.sum(): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of all elements in the
array.\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Array<out Int>.sum(): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of all elements in
the array.\n */\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Array<out Long>.sum(): Long {\n    var sum:
Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of
all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Array<out Float>.sum(): Float
{\n    var sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Array<out Double>.sum(): Double {\n    var sum:
Double = 0.0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/* Returns the sum of

```



```

IntRange.randomOrNull(random:
Random): Int? {\n if (isEmpty())\n return null\n return random.nextInt(this)\n}\n\n/**\n * Returns a random
element from this range using the specified source of randomness, or `null` if this range is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongRange.randomOrNull(random: Random): Long? {\n if (isEmpty())\n return null\n return
random.nextLong(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharRange.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return
random.nextInt(first.code, last.code + 1).toChar()\n}\n\n/**\n * Returns `true` if this range contains the specified
[element].\n * \n * Always returns `false` if the [element] is `null`.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline operator fun IntRange.contains(element: Int?): Boolean {\n return element != null &&
contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns
`false` if the [element] is `null`.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator
fun LongRange.contains(element: Long?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n *
Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is
`null`.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun
CharRange.contains(element: Char?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("intRangeContains")\npublic
operator fun ClosedRange<Int>.contains(value: Byte): Boolean {\n return contains(value.toInt())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value:
Byte): Boolean {\n return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun
ClosedRange<Short>.contains(value: Byte): Boolean {\n return contains(value.toShort())\n}\n\n/**\n * Checks if
the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and
floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Byte): Boolean {\n return contains(value.toDouble())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This
`contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value:
Byte): Boolean {\n return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous
semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4",
hiddenSince = "1.5")\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Double): Boolean {\n return value.toIntExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@Deprecated("This `contains` operation
mixing integer and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value:
Double): Boolean {\n return value.toLongExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be

```



```

\"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value:
Int): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value:
Long): Boolean {\n    return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic
operator fun ClosedRange<Byte>.contains(value: Long): Boolean {\n    return value.toByteExactOrNull().let { if (it
!= null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun ClosedRange<Short>.contains(value:
Long): Boolean {\n    return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince =
\"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun
ClosedRange<Double>.contains(value: Long): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince
= \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun
ClosedRange<Float>.contains(value: Long): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Checks if
the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun
ClosedRange<Int>.contains(value: Short): Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun
ClosedRange<Long>.contains(value: Short): Boolean {\n    return contains(value.toLong())\n}\n\n/**\n * Checks if
the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator
fun ClosedRange<Byte>.contains(value:
Short): Boolean {\n    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince =
\"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun
ClosedRange<Double>.contains(value: Short): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince =
\"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value:
Short): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Returns a progression from this value down to
the specified [to] value with the step -1.\n */\n\n * The [to] value should be less than or equal to `this` value.\n *
If the [to] value is greater than `this` value the returned progression is empty.\n */\n\npublic infix fun Int.downTo(to: Byte):
IntProgression {\n    return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/**\n * Returns a progression
from this value down to the specified [to] value with the step -1.\n */\n\n * The [to] value should be less than or equal
to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n\npublic infix
fun Long.downTo(to: Byte): LongProgression {\n    return LongProgression.fromClosedRange(this, to.toLong(), -
1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n */\n\n *
The [to] value should be less than or equal to `this`
value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n\npublic infix fun
Byte.downTo(to: Byte): IntProgression {\n    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -
1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n */\n\n *
The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned

```



```

1)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n
*\npublic fun IntProgression.reversed(): IntProgression {\n    return IntProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction
with the same step.\n *\npublic fun LongProgression.reversed(): LongProgression {\n    return
LongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same
range in the opposite direction with the same step.\n *\npublic fun CharProgression.reversed(): CharProgression {\n
    return CharProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the
same range with the given step.\n *\npublic infix fun IntProgression.step(step: Int): IntProgression {\n
    checkStepIsPositive(step > 0, step)\n    return IntProgression.fromClosedRange(first, last, if (this.step > 0) step else -
step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n *\npublic infix fun
LongProgression.step(step: Long): LongProgression {\n    checkStepIsPositive(step > 0, step)\n    return
LongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a
progression that goes over the same range with the given step.\n *\npublic infix fun CharProgression.step(step:
Int): CharProgression {\n    checkStepIsPositive(step > 0, step)\n    return CharProgression.fromClosedRange(first,
last, if (this.step > 0) step else -step)\n}\n\ninternal fun Int.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toInt()..Byte.MAX_VALUE.toInt()) this.toByte() else null\n}\n\ninternal fun
Long.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toLong()..Byte.MAX_VALUE.toLong()) this.toByte() else null\n}\n\ninternal fun
Short.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort()) this.toByte() else null\n}\n\ninternal fun
Double.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toInt().toByte() else null\n}\n\ninternal fun
Float.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat())
    this.toInt().toByte() else null\n}\n\ninternal fun Long.toIntExactOrNull(): Int? {\n    return if (this in
Int.MIN_VALUE.toLong()..Int.MAX_VALUE.toLong()) this.toInt() else null\n}\n\ninternal fun
Double.toIntExactOrNull(): Int? {\n    return if (this in
Int.MIN_VALUE.toDouble()..Int.MAX_VALUE.toDouble()) this.toInt() else null\n}\n\ninternal fun
Float.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toFloat()..Int.MAX_VALUE.toFloat())
    this.toInt() else null\n}\n\ninternal fun Double.toLongExactOrNull(): Long? {\n    return if (this in
Long.MIN_VALUE.toDouble()..Long.MAX_VALUE.toDouble()) this.toLong() else null\n}\n\ninternal fun
Float.toLongExactOrNull(): Long? {\n    return if (this in
Long.MIN_VALUE.toFloat()..Long.MAX_VALUE.toFloat()) this.toLong() else null\n}\n\ninternal fun
Int.toShortExactOrNull(): Short? {\n    return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())
    this.toShort() else null\n}\n\ninternal fun Long.toShortExactOrNull():
    Short? {\n    return if (this in Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong()) this.toShort() else
    null\n}\n\ninternal fun Double.toShortExactOrNull(): Short? {\n    return if (this in
Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble()) this.toInt().toShort() else null\n}\n\ninternal fun
Float.toShortExactOrNull(): Short? {\n    return if (this in
Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat()) this.toInt().toShort() else null\n}\n\n/**\n * Returns
a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to
`this` value, then the returned range is empty.\n *\npublic infix fun Int.until(to: Byte): IntRange {\n    return this ..
(to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n *
\n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\npublic infix fun
Long.until(to: Byte): LongRange {\n
    return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the
specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*\npublic infix fun Byte.until(to: Byte): IntRange {\n    return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n *

```


Byte { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Short.coerceAtLeast(minimumValue: Short):

Short { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample

samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Int.coerceAtLeast(minimumValue: Int): Int { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Long.coerceAtLeast(minimumValue: Long): Long { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue]

or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Float.coerceAtLeast(minimumValue: Float): Float { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun

Double.coerceAtLeast(minimumValue: Double): Double { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostComparable\n */\npublic fun <T : Comparable<T>>

T.coerceAtMost(maximumValue: T): T { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun

Byte.coerceAtMost(maximumValue: Byte): Byte { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Short.coerceAtMost(maximumValue: Short):

Short { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun

Int.coerceAtMost(maximumValue: Int): Int { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Long.coerceAtMost(maximumValue: Long):

Long { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Float.coerceAtMost(maximumValue: Float):

Float { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Double.coerceAtMost(maximumValue:

Double): Double { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this

```

value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range,
or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n */\npublic fun
<T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n    if (minimumValue !== null
&& maximumValue !== null) {\n        if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n        if (this < minimumValue) return minimumValue\n        if (this > maximumValue) return maximumValue\n    }
else {\n        if (minimumValue !== null && this < minimumValue) return minimumValue\n        if
(maximumValue !== null && this > maximumValue) return maximumValue\n    }\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is
greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun
Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {\n    if (minimumValue > maximumValue)
throw IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Short.coerceIn(minimumValue: Short,
maximumValue: Short): Short {\n    if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Int.coerceIn(minimumValue: Int,
maximumValue: Int): Int {\n    if (minimumValue > maximumValue) throw IllegalArgumentOutOfRangeException("Cannot
coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this <
minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return
this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Long.coerceIn(minimumValue: Long,
maximumValue: Long): Long {\n    if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Float.coerceIn(minimumValue: Float,
maximumValue: Float): Float {\n    if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this

```

```

value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n *
@sample samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Double.coerceIn(minimumValue:
Double, maximumValue: Double): Double {\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue)\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInFloatingPointRange\n *^\n@SinceKotlin("1.1")\npublic fun <T :
Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {\n if (range.isEmpty())
throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")\n return when {\n //
this < start equiv to this <= start && !(this >= start)\n range.lessThanOrEqualTo(this, range.start) &&
!range.lessThanOrEqualTo(range.start, this) -> range.start\n // this > end equiv to this >= end && !(this <= end)\n
range.lessThanOrEqualTo(range.endInclusive, this) && !range.lessThanOrEqualTo(this, range.endInclusive) ->
range.endInclusive\n else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInComparable\n *^\npublic fun <T : Comparable<T>>
T.coerceIn(range: ClosedRange<T>): T {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<T>(range)\n
}\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")\n
return when {\n this < range.start -> range.start\n this > range.endInclusive -> range.endInclusive\n
else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's
in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater
than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun
Int.coerceIn(range: ClosedRange<Int>): Int {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<Int>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to
an empty range: $range.")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -
> range.endInclusive\n else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start`
if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n *
@sample samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Long.coerceIn(range:
ClosedRange<Long>): Long {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<Long>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value
to an empty range: $range.")\n return when {\n this < range.start -> range.start\n this >
range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n"/\n * Copyright 2010-2022 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *^\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage
kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npub
lic value class UByte @PublishedApi internal constructor(@PublishedApi internal val data: Byte) :
Comparable<UByte> {\n\n companion object {\n /**\n * A constant holding the minimum value an
instance of UByte can have.\n *^\n public const val MIN_VALUE: UByte = UByte(0)\n\n /**\n *
A constant holding the maximum value an instance of UByte can have.\n *^\n public const val
MAX_VALUE: UByte = UByte(-1)\n\n /**\n * The number of bytes used to represent an instance of
UByte in a binary form.\n *^\n public const val SIZE_BYTES: Int = 1\n\n /**\n * The number of
bits used to represent an instance of UByte in a binary form.\n *^\n public const val SIZE_BITS: Int = 8\n

```

```

}\n\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n
@Suppress("OVERRIDE_BY_INLINE")\n public override inline operator fun compareTo(other: UByte): Int =
this.toInt().compareTo(other.toInt())\n\n /**\n * Compares this value with the specified value for order.\n *
Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a
positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())\n\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive
number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n /** Adds the
other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): UInt =
this.toUInt().plus(other.toInt())\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())\n\n /** Adds the other value
to this value. */\n @kotlin.internal.InlineOnly\n public
inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)\n\n /** Adds the other value to this value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong =
this.toULong().plus(other)\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n\n /** Subtracts the
other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())\n\n /** Subtracts the other value from this value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n\n
/** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun
minus(other: ULong): ULong = this.toULong().minus(other)\n\n /** Multiplies this value by
the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toInt())\n\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())\n\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n\n
/** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun
times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n\n /** Divides this value by the other value,
truncating the result to an integer
that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UShort): UInt =
this.toUInt().div(other.toUInt())\n\n /** Divides this value by the other value, truncating the result to an integer that
is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UInt): UInt =
this.toUInt().div(other)\n\n /** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())\n\n /**\n * Calculates the
remainder of truncating division of this value by the other

```



```

value.\n * \n * The result is always less than the divisor.\n * \n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt =
this.toUInt().rem(other)\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n * \n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types,
the results of flooring division and truncating division are
the same.\n * \n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): UInt =
this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an
integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and
truncating division are the same.\n * \n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring
the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring
division and truncating division are the same.\n * \n @kotlin.internal.InlineOnly\n public inline fun
floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n * Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.\n
* \n * For unsigned types, the results of flooring division and truncating division are the same.\n * \n
@kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong =
this.toULong().floorDiv(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n * \n @kotlin.internal.InlineOnly\n public inline
fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder
of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n
* For unsigned types, the remainders of flooring division and truncating division are the same.\n * \n
@kotlin.internal.InlineOnly\n public inline fun mod(other: UShort):
UShort = this.toUInt().mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder of flooring division
of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned
types, the remainders of flooring division and truncating division are the same.\n * \n
@kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n /**\n *
Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less
than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the
same.\n * \n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong =
this.toULong().mod(other)\n\n /**\n * Returns this value incremented by one.\n * \n * @sample
samples.misc.Builtins.inc\n * \n @kotlin.internal.InlineOnly\n public inline operator
fun inc(): UByte = UByte(data.inc())\n\n /**\n * Returns this value decremented by one.\n * \n * @sample
samples.misc.Builtins.dec\n * \n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UByte =
UByte(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. *\n
@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UByte): UIntRange =
UIntRange(this.toUInt(), other.toUInt())\n\n /**\n * Performs a bitwise AND operation between the two values. *\n
@kotlin.internal.InlineOnly\n public inline infix fun and(other: UByte): UByte = UByte(this.data and other.data)\n
/**\n * Performs a bitwise OR operation between the two values. *\n @kotlin.internal.InlineOnly\n public inline
infix fun or(other: UByte): UByte = UByte(this.data or other.data)\n\n /**\n * Performs a bitwise XOR operation
between the two values. *\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UByte):
UByte = UByte(this.data xor other.data)\n\n /**\n * Inverts the bits in this value. *\n @kotlin.internal.InlineOnly\n
public inline fun inv(): UByte = UByte(data.inv())\n\n /**\n * Converts this [UByte] value to [Byte].\n * \n
* If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same

```

```

numerical value as this `UByte`. Otherwise the result is negative.
    * The resulting `Byte` value has the
    same binary representation as this `UByte` value.
    @kotlin.internal.InlineOnly
    public inline fun
    toByte(): Byte = data
    /**
    * Converts this [UByte] value to [Short].
    * The resulting `Short` value
    represents the same numerical value as this `UByte`.
    * The least significant 8 bits of the resulting `Short`
    value are the same as the bits of this `UByte` value,
    * whereas the most significant 8 bits are filled with zeros.
    */
    @kotlin.internal.InlineOnly
    public inline fun toShort(): Short = data.toShort() and 0xFF
    /**
    * Converts this [UByte] value to [Int].
    * The resulting `Int` value represents the same numerical value as this `UByte`.
    * The least
    significant 8 bits of the resulting `Int` value are the same as the bits of this `UByte` value,
    * whereas the most
    significant 24 bits are filled with zeros.
    */
    @kotlin.internal.InlineOnly
    public inline fun toInt(): Int =
    data.toInt() and 0xFF
    /**
    * Converts this [UByte] value to [Long].
    * The resulting `Long` value
    represents the same numerical value as this `UByte`.
    * The least significant 8 bits of the resulting `Long`
    value are the same as the bits of this `UByte` value,
    * whereas the most significant 56 bits are filled with
    zeros.
    */
    @kotlin.internal.InlineOnly
    public inline fun toLong(): Long = data.toLong() and 0xFF
    /** Returns this value. */
    @kotlin.internal.InlineOnly
    public inline fun toUByte(): UByte = this
    /**
    * Converts this [UByte]
    value to [UShort].
    * The resulting `UShort` value represents the same numerical value as this `UByte`.
    * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `UByte` value,
    * whereas the most significant 8 bits are filled with zeros.
    */
    @kotlin.internal.InlineOnly
    public inline
    fun toUShort(): UShort = UShort(data.toShort() and 0xFF)
    /**
    * Converts this [UByte] value to [UInt].
    * The resulting `UInt` value represents the same numerical value as this `UByte`.
    * The least
    significant 8 bits of the resulting `UInt` value are the same as the bits of this `UByte` value,
    * whereas the most
    significant 24 bits are filled with zeros.
    */
    @kotlin.internal.InlineOnly
    public inline fun toUInt(): UInt =
    UInt(data.toInt() and 0xFF)
    /**
    * Converts this [UByte] value to [ULong].
    * The resulting `ULong` value represents the same
    numerical value as this `UByte`.
    * The least significant 8 bits of the resulting `ULong` value are the same
    as the bits of this `UByte` value,
    * whereas the most significant 56 bits are filled with zeros.
    */
    @kotlin.internal.InlineOnly
    public inline fun toULong(): ULong = ULong(data.toLong() and 0xFF)
    /**
    * Converts this [UByte] value to [Float].
    * The resulting `Float` value represents the same numerical
    value as this `UByte`.
    */
    @kotlin.internal.InlineOnly
    public inline fun toFloat(): Float =
    this.toInt().toFloat()
    /**
    * Converts this [UByte] value to [Double].
    * The resulting `Double`
    value represents the same numerical value as this `UByte`.
    */
    @kotlin.internal.InlineOnly
    public inline
    fun toDouble(): Double = this.toInt().toDouble()
    public
    override fun toString(): String = toInt().toString()
    /**
    * Converts this [Byte] value to [UByte].
    * If
    this value is positive, the resulting `UByte` value represents the same numerical value as this `Byte`.
    * The
    resulting `UByte` value has the same binary representation as this `Byte` value.
    */
    @SinceKotlin("1.5")
    @WasExperimental(ExperimentalUnsignedTypes::class)
    @kotlin.internal.InlineOnly
    public inline fun Byte.toUByte(): UByte = UByte(this)
    /**
    * Converts this [Short] value to [UByte].
    * If
    this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents
    * the same numerical value as this `Short`.
    * The resulting `UByte` value is represented by the least significant 8
    bits of this `Short` value.
    */
    @SinceKotlin("1.5")
    @WasExperimental(ExperimentalUnsignedTypes::class)
    @kotlin.internal.InlineOnly
    public inline fun Short.toUByte(): UByte = UByte(this.toByte())
    /**
    * Converts this [Int]
    value to [UByte].
    * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting
    `UByte` value represents
    * the same numerical value as this `Int`.
    * The resulting `UByte` value is
    represented by the least significant 8 bits of this `Int` value.
    */
    @SinceKotlin("1.5")
    @WasExperimental(ExperimentalUnsignedTypes::class)
    @kotlin.internal.InlineOnly
    public inline fun Int.toUByte(): UByte = UByte(this.toByte())
    /**
    * Converts this [Long] value to [UByte].
    */

```

```

*\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value
represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UByte` value is represented by the
least significant 8 bits of this `Long` value.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Long.toUByte(): UByte = UByte(this.toByte())\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o.
and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @JvmInline\n pu
blic value class UInt @PublishedApi internal constructor(@PublishedApi internal val data: Int) :
Comparable<UInt> {\n\n    companion object {\n\n        /**\n         * A constant holding the minimum value an
instance of UInt can have.\n         *\n         * public const val MIN_VALUE: UInt = UInt(0)\n\n         */\n         * A
constant holding the maximum value an instance of UInt can have.\n         *\n         * public const val MAX_VALUE:
UInt = UInt(-1)\n\n         */\n         * The number of bytes used to represent an instance of UInt in a binary form.\n
         *\n         * public const val SIZE_BYTES: Int = 4\n\n         */\n         * The number of bits used to represent an instance of UInt in a binary form.\n         *\n         * public const
val SIZE_BITS: Int = 32\n    }\n\n    /**\n     * Compares this value with the specified value for order.\n     * Returns
zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive
number if it's greater than other.\n     *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun
compareTo(other: UByte): Int = this.compareTo(other.toUInt())\n\n     */\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater than other.\n     *\n     * @kotlin.internal.InlineOnly\n
     * public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toUInt())\n\n     */\n     *
Compares this value with the specified value for order.\n
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or
a positive number if it's greater than other.\n     *\n     * @kotlin.internal.InlineOnly\n
     * @Suppress("OVERRIDE_BY_INLINE")\n     * public override inline operator fun compareTo(other: UInt): Int =
uintCompare(this.data, other.data)\n\n     */\n     * Compares this value with the specified value for order.\n     *
Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a
positive number if it's greater than other.\n     *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun
compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n     */\n     * Adds the other value to this value. *\n
     * @kotlin.internal.InlineOnly\n     * public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())\n
     */\n     * Adds the other value to this value. *\n     * @kotlin.internal.InlineOnly\n
     * public inline operator fun plus(other: UShort): UInt = this.plus(other.toUInt())\n\n     */\n     * Adds the other value to this
value. *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun plus(other: UInt): UInt =
UInt(this.data.plus(other.data))\n\n     */\n     * Adds the other value to this value. *\n     * @kotlin.internal.InlineOnly\n
     * public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n\n     */\n     * Subtracts the other
value from this value. *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun minus(other: UByte): UInt =
this.minus(other.toUInt())\n\n     */\n     * Subtracts the other value from this value. *\n     * @kotlin.internal.InlineOnly\n
     * public inline operator fun minus(other: UShort): UInt = this.minus(other.toUInt())\n\n     */\n     * Subtracts the other value
from this value. *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun minus(other: UInt): UInt =
UInt(this.data.minus(other.data))\n\n     */\n     * Subtracts the other value from
this value. *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun minus(other: ULong): ULong =
this.toULong().minus(other)\n\n     */\n     * Multiplies this value by the other value. *\n     * @kotlin.internal.InlineOnly\n
     * public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())\n\n     */\n     * Multiplies this value by the
other value. *\n     * @kotlin.internal.InlineOnly\n     * public inline operator fun times(other: UShort): UInt =
this.times(other.toUInt())\n\n     */\n     * Multiplies this value by the other value. *\n     * @kotlin.internal.InlineOnly\n

```

```

public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))\n /** Multiplies this value
by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong =
this.toULong().times(other)\n\n /** Divides this value by the other value, truncating the result to an integer that is
closer to zero. *\n @kotlin.internal.InlineOnly\n public
inline operator fun div(other: UByte): UInt = this.div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UShort): UInt = this.div(other.toUInt())\n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other:
UInt): UInt = uintDivide(this, other)\n /** Divides this value by the other value, truncating the result to an integer
that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n *\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other:
UByte): UInt = this.rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value
by the other value.\n * \n * The result is always less than the divisor.\n *\n @kotlin.internal.InlineOnly\n
public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n *\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt =
uintRemainder(this, other)\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n *\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by
the other value, flooring the result to an integer that is
closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division
are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): UInt =
this.floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that
is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division
are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UShort): UInt =
this.floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that
is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division
are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): UInt = div(other)\n
/**\n * Divides this value
by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
*\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte =
this.mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n
*\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort =
this.mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline
fun mod(other: UInt): UInt = rem(other)\n /**\n * Calculates the remainder of flooring division of this value by
the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value

```

```

incremented by one.\n * \n * @sample samples.misc.Builtins.inc\n * \n
@kotlin.internal.InlineOnly\n public inline operator fun inc(): UInt = UInt(data.inc())\n\n /**\n * Returns
this value decremented by one.\n * \n * @sample samples.misc.Builtins.dec\n * \n
@kotlin.internal.InlineOnly\n public inline operator fun dec(): UInt = UInt(data.dec())\n\n /** Creates a range
from this value to the specified [other] value. *\n @kotlin.internal.InlineOnly\n public inline operator fun
rangeTo(other: UInt): UIntRange = UIntRange(this, other)\n\n /**\n * Shifts this value left by the [bitCount]
number of bits.\n * \n * Note that only the five lowest-order bits of the [bitCount] are used as the shift
distance.\n * The shift distance actually used is therefore always in the range `0..31`.\n * \n
@kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount: Int): UInt = UInt(data shl bitCount)\n\n /**\n
* Shifts this value right by the [bitCount] number of bits, filling the leftmost bits
with zeros.\n * \n * Note that only the five lowest-order bits of the [bitCount] are used as the shift distance.\n
* The shift distance actually used is therefore always in the range `0..31`.\n * \n @kotlin.internal.InlineOnly\n
public inline infix fun shr(bitCount: Int): UInt = UInt(data ushr bitCount)\n\n /** Performs a bitwise AND
operation between the two values. *\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UInt):
UInt = UInt(this.data and other.data)\n\n /** Performs a bitwise OR operation between the two values. *\n
@kotlin.internal.InlineOnly\n public inline infix fun or(other: UInt): UInt = UInt(this.data or other.data)\n\n /**
Performs a bitwise XOR operation between the two values. *\n @kotlin.internal.InlineOnly\n public inline infix
fun xor(other: UInt): UInt = UInt(this.data xor other.data)\n\n /** Inverts the bits in this value. *\n
@kotlin.internal.InlineOnly\n public inline fun inv(): UInt
= UInt(data.inv())\n\n /**\n * Converts this [UInt] value to [Byte].\n * \n * If this value is less than or
equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this
`UInt`.\n * \n * The resulting `Byte` value is represented by the least significant 8 bits of this `UInt` value.\n *
Note that the resulting `Byte` value may be negative.\n * \n @kotlin.internal.InlineOnly\n public inline fun
toByte(): Byte = data.toByte()\n\n /**\n * Converts this [UInt] value to [Short].\n * \n * If this value is less
than or equals to [Short.MAX_VALUE], the resulting `Short` value represents\n * the same numerical value as
this `UInt`.\n * \n * The resulting `Short` value is represented by the least significant 16 bits of this `UInt`
value.\n * Note that the resulting `Short` value may be negative.\n * \n @kotlin.internal.InlineOnly\n public
inline fun toShort(): Short = data.toShort()\n\n
/**\n * Converts this [UInt] value to [Int].\n * \n * If this value is less than or equals to
[Int.MAX_VALUE], the resulting `Int` value represents\n * the same numerical value as this `UInt`. Otherwise
the result is negative.\n * \n * The resulting `Int` value has the same binary representation as this `UInt` value.\n
* \n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data\n\n /**\n * Converts this [UInt] value
to [Long].\n * \n * The resulting `Long` value represents the same numerical value as this `UInt`.\n * \n *
The least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt` value,\n *
whereas the most significant 32 bits are filled with zeros.\n * \n @kotlin.internal.InlineOnly\n public inline fun
toLong(): Long = data.toLong() and 0xFFFF_FFFF\n\n /**\n * Converts this [UInt] value to [UByte].\n * \n
* If this value is less than or equals to
[UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `UInt`.\n
*\n * The resulting `UByte` value is represented by the least significant 8 bits of this `UInt` value.\n * \n
@kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = data.toUByte()\n\n /**\n * Converts this
[UInt] value to [UShort].\n * \n * If this value is less than or equals to [UShort.MAX_VALUE], the resulting
`UShort` value represents\n * the same numerical value as this `UInt`.\n * \n * The resulting `UShort` value is
represented by the least significant 16 bits of this `UInt` value.\n * \n @kotlin.internal.InlineOnly\n public
inline fun toUShort(): UShort = data.toUShort()\n\n /** Returns this value. *\n @kotlin.internal.InlineOnly\n
public inline fun toUInt(): UInt = this\n\n /**\n * Converts this [UInt] value to [ULong].\n * \n * The
resulting `ULong` value represents the same numerical value
as this `UInt`.\n * \n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of
this `UInt` value,\n * whereas the most significant 32 bits are filled with zeros.\n * \n

```

```

@kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong() and
0xFFFF_FFFF)\n\n /**\n * Converts this [UInt] value to [Float].\n * The resulting value is the closest
`Float` to this `UInt` value.\n * In case when this `UInt` value is exactly between two `Float`s,\n * the one with
zero at least significant bit of mantissa is selected.\n */\n @kotlin.internal.InlineOnly\n public inline fun
toFloat(): Float = this.toDouble().toFloat()\n\n /**\n * Converts this [UInt] value to [Double].\n * The
resulting `Double` value represents the same numerical value as this `UInt`.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = uintToDouble(data)\n\n public
override fun toString(): String = toLong().toString()\n\n}\n\n/**\n * Converts this [Byte] value to [UInt].\n */\n * If
this value is positive, the resulting `UInt` value represents the same numerical value as this `Byte`.\n * The least
significant 8 bits of the resulting `UInt` value are the same as the bits of this `Byte` value,\n * whereas the most
significant 24 bits are filled with the sign bit of this value.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Byte.toUInt(): UInt = UInt(this.toInt())\n\n /**\n * Converts this [Short] value to [UInt].\n */\n * If
this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`.\n * The least
significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value,\n * whereas the most
significant 16 bits are filled with the sign bit of this value.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Short.toUInt(): UInt = UInt(this.toInt())\n\n /**\n * Converts this [Int] value to [UInt].\n */\n * If this value
is positive, the resulting `UInt` value represents the same numerical value as this `Int`.\n * The resulting `UInt`
value has the same binary representation as this `Int` value.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Int.toUInt(): UInt = UInt(this)\n\n /**\n * Converts this [Long] value to [UInt].\n */\n * If this value
is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents\n * the same
numerical value as this `Long`.\n * The resulting `UInt` value is represented by the least significant 32 bits of
this `Long` value.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Long.toUInt(): UInt =
    UInt(this.toInt())\n\n /**\n * Converts this [Float] value to [UInt].\n */\n * The fractional part, if any, is rounded
down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger
than `UInt.MAX_VALUE`.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Float.toUInt(): UInt = doubleToUInt(this.toDouble())\n\n /**\n * Converts this [Double] value to
[UInt].\n */\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Double` value is
negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.\n
*/\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Double.toUInt(): UInt = doubleToUInt(this)\n\n", "\n\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @JvmInline\n public
value class UShort @PublishedApi internal constructor(@PublishedApi internal val data: Short) :
Comparable<UShort> {\n\n companion object {\n\n /**\n * A constant holding the minimum value an
instance of UShort can have.\n */\n public const val MIN_VALUE: UShort = UShort(0)\n\n /**\n
* A constant holding the maximum value an instance of UShort can have.\n */\n public const val
MAX_VALUE: UShort = UShort(-1)\n\n /**\n * The number of bytes used to represent an instance of
UShort in a binary form.\n */\n public const val SIZE_BYTES: Int = 2\n\n /**\n * The number of
bits used to represent an instance of UShort in a binary form.\n
*/\n

```

```

    *^n    public const val SIZE_BITS: Int = 16^n    }^n    /**^n    * Compares this value with the specified
value for order.^n    * Returns zero if this value is equal to the specified other value, a negative number if it's less
than other.^n    * or a positive number if it's greater than other.^n    *^n    @kotlin.internal.InlineOnly^n    public
inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())^n    /**^n    * Compares
this value with the specified value for order.^n    * Returns zero if this value is equal to the specified other value, a
negative number if it's less than other.^n    * or a positive number if it's greater than other.^n    *^n
@kotlin.internal.InlineOnly^n    @Suppress("OVERRIDE_BY_INLINE")^n    public override inline operator fun
compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())^n    /**^n    * Compares this value with the
specified value for order.^n    * Returns zero if
this value is equal to the specified other value, a negative number if it's less than other.^n    * or a positive number
if it's greater than other.^n    *^n    @kotlin.internal.InlineOnly^n    public inline operator fun compareTo(other:
UInt): Int = this.toUInt().compareTo(other)^n    /**^n    * Compares this value with the specified value for order.^n
    * Returns zero if this value is equal to the specified other value, a negative number if it's less than other.^n    * or a
positive number if it's greater than other.^n    *^n    @kotlin.internal.InlineOnly^n    public inline operator fun
compareTo(other: ULong): Int = this.toULong().compareTo(other)^n    /** Adds the other value to this value. *^n
    @kotlin.internal.InlineOnly^n    public inline operator fun plus(other: UByte): UInt =
this.toUInt().plus(other.toInt())^n    /** Adds the other value to this value. *^n    @kotlin.internal.InlineOnly^n
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())^n
    /** Adds the other value to this value. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun plus(other:
UInt): UInt = this.toUInt().plus(other)^n    /** Adds the other value to this value. *^n
    @kotlin.internal.InlineOnly^n    public inline operator fun plus(other: ULong): ULong =
this.toULong().plus(other)^n    /** Subtracts the other value from this value. *^n    @kotlin.internal.InlineOnly^n
public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())^n    /** Subtracts the
other value from this value. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())^n    /** Subtracts the other value from this value. *^n
    @kotlin.internal.InlineOnly^n    public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)^n
    /** Subtracts the other value from this value. *^n    @kotlin.internal.InlineOnly^n    public
inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)^n    /** Multiplies this value by
the other value. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toInt())^n    /** Multiplies this value by the other value. *^n
    @kotlin.internal.InlineOnly^n    public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())^n    /** Multiplies this value by the other value. *^n
    @kotlin.internal.InlineOnly^n    public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)^n
    /** Multiplies this value by the other value. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun
times(other: ULong): ULong = this.toULong().times(other)^n    /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *^n    @kotlin.internal.InlineOnly^n    public inline operator
fun div(other: UByte): UInt =
this.toUInt().div(other.toUInt())^n    /** Divides this value by the other value, truncating the result to an integer that
is closer to zero. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun div(other: UShort): UInt =
this.toUInt().div(other.toUInt())^n    /** Divides this value by the other value, truncating the result to an integer that
is closer to zero. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun div(other: UInt): UInt =
this.toUInt().div(other)^n    /** Divides this value by the other value, truncating the result to an integer that is closer
to zero. *^n    @kotlin.internal.InlineOnly^n    public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)^n    /**^n    * Calculates the remainder of truncating division of this value by the other
value.^n    *^n    * The result is always less than the divisor.^n    *^n    @kotlin.internal.InlineOnly^n    public
inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())^n
    /**^n    * Calculates the remainder of truncating division of this value by the other value.^n    *^n    * The result
is always less than the divisor.^n    *^n    @kotlin.internal.InlineOnly^n    public inline operator fun rem(other:

```

```

UShort): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of
this value by the other value.\n * \n * The result is always less than the divisor.\n */\n
@kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)\n /**\n
* Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always
less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong):
ULong = this.toULong().rem(other)\n /**\n * Divides this value by the other value, flooring the result to an
integer that is closer to negative
infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n
*/\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): UInt =
this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an
integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and
truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring
the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring
division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun
floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n
* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n
* For unsigned types, the results of flooring division and truncating division are the same.\n */\n
@kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong =
this.toULong().floorDiv(other)\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline
fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder
of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n
* For unsigned types, the remainders of flooring division and truncating
division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort =
this.toUInt().mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n /**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n
* For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n
@kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n
/**\n * Returns this value incremented by one.\n
*\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
inc(): UShort = UShort(data.inc())\n /**\n * Returns this value decremented by one.\n * \n * @sample
samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UShort =
UShort(data.dec())\n /**\n * Creates a range from this value to the specified [other] value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UShort): UIntRange =
UIntRange(this.toUInt(), other.toUInt())\n /**\n * Performs a bitwise AND operation between the two values. */\n
@kotlin.internal.InlineOnly\n public inline infix fun and(other: UShort): UShort = UShort(this.data and
other.data)\n /**\n * Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n
public inline infix fun or(other: UShort): UShort = UShort(this.data or other.data)\n /**\n * Performs a
bitwise XOR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun
xor(other: UShort): UShort = UShort(this.data xor other.data)\n /**\n * Inverts the bits in this value. */\n
@kotlin.internal.InlineOnly\n public inline fun inv(): UShort = UShort(data.inv())\n /**\n * Converts this
[UShort] value to [Byte].\n * \n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting
`Byte` value represents\n * the same numerical value as this `UShort`.\n * \n * The resulting `Byte` value is

```


represented by the least significant 8 bits of this `UShort` value.
 * Note that the resulting `Byte` value may be negative.
 @kotlin.internal.InlineOnly
 public inline fun toByte(): Byte = data.toByte()
 /**
 * Converts this [UShort] value to [Short].
 * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents the same numerical value as this `UShort`. Otherwise the result is negative.
 * The resulting `Short` value has the same binary representation as this `UShort` value.
 @kotlin.internal.InlineOnly
 public inline fun toShort(): Short = data
 /**
 * Converts this [UShort] value to [Int].
 * The resulting `Int` value represents the same numerical value as this `UShort`.
 * The least significant 16 bits of the resulting `Int` value are the same as the bits of this `UShort` value, whereas the most significant 16 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline fun toInt(): Int = data.toInt() and 0xFFFF
 /**
 * Converts this [UShort] value to [Long].
 * The resulting `Long` value represents the same numerical value as this `UShort`.
 * The least significant 16 bits of the resulting `Long` value are the same as the bits of this `UShort` value, whereas the most significant 48 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline fun toLong(): Long = data.toLong() and 0xFFFF
 /**
 * Converts this [UShort] value to [UByte].
 * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `UShort`.
 * The resulting `UByte` value is represented by the least significant 8 bits of this `UShort` value.
 @kotlin.internal.InlineOnly
 public inline fun toUByte(): UByte = data.toUByte()
 /**
 * Returns this value.
 @kotlin.internal.InlineOnly
 public inline fun toUShort(): UShort = this
 /**
 * Converts this [UShort] value to [UInt].
 * The resulting `UInt` value represents the same numerical value as this `UShort`.
 * The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `UShort` value, whereas the most significant 16 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF)
 /**
 * Converts this [UShort] value to [ULong].
 * The resulting `ULong` value represents the same numerical value as this `UShort`.
 * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `UShort` value, whereas the most significant 48 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF)
 /**
 * Converts this [UShort] value to [Float].
 * The resulting `Float` value represents the same numerical value as this `UShort`.
 @kotlin.internal.InlineOnly
 public inline fun toFloat(): Float = this.toInt().toFloat()
 /**
 * Converts this [UShort] value to [Double].
 * The resulting `Double` value represents the same numerical value as this `UShort`.
 @kotlin.internal.InlineOnly
 public inline fun toDouble(): Double = this.toInt().toDouble()
 public override fun toString(): String = toInt().toString()
 /**
 * Converts this [Byte] value to [UShort].
 * If this value is positive, the resulting `UShort` value represents the same numerical value as this `Byte`.
 * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `Byte` value, whereas the most significant 8 bits are filled with the sign bit of this value.
 @SinceKotlin("1.5")
 @WasExperimental(ExperimentalUnsignedTypes::class)
 @kotlin.internal.InlineOnly
 public inline fun Byte.toUShort(): UShort = UShort(this.toShort())
 /**
 * Converts this [Short] value to [UShort].
 * If this value is positive, the resulting `UShort` value represents the same numerical value as this `Short`.
 * The resulting `UShort` value has the same binary representation as this `Short` value.
 @SinceKotlin("1.5")
 @WasExperimental(ExperimentalUnsignedTypes::class)
 @kotlin.internal.InlineOnly
 public inline fun Short.toUShort(): UShort = UShort(this)
 /**
 * Converts this [Int] value to [UShort].
 * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same numerical value as this `Int`.
 * The resulting `UShort` value is represented by the least significant 16 bits of this `Int` value.
 @SinceKotlin("1.5")
 @WasExperimental(ExperimentalUnsignedTypes::class)
 @kotlin.internal.InlineOnly

```

public inline fun Int.toUShort(): UShort = UShort(this.toShort())\n/**\n * Converts this [Long] value to
[UShort].\n *\n * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort`
value represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UShort` value is represented by
the least significant 16
bits of this `Long` value.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly
public inline fun Long.toUShort(): UShort = UShort(this.toShort())\n"/*\n * Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin.ranges\n\n/**\n * A range of values of type `Char`.\n */\npublic class CharRange(start: Char, endInclusive:
Char) : CharProgression(start, endInclusive, 1), ClosedRange<Char> {\n    override val start: Char get() = first\n
    override val endInclusive: Char get() = last\n\n    override fun contains(value: Char): Boolean = first <= value &&
value <= last\n\n    /**\n     * Checks whether the range is empty.\n     *\n     * The range is empty if its start value is
greater than the end value.\n     */\n    override fun
isEmpty(): Boolean = first > last\n\n    override fun equals(other: Any?): Boolean =\n        other is CharRange &&
(isEmpty() && other.isEmpty()) ||\n        first == other.first && last == other.last)\n\n    override fun hashCode(): Int
=\n        if (isEmpty()) -1 else (31 * first.code + last.code)\n\n    override fun toString(): String = "$first..$last"\n\n
companion object {\n        /** An empty range of values of type Char. */\n        public val EMPTY: CharRange =
CharRange(1.toChar(), 0.toChar())\n    }\n}\n\n/**\n * A range of values of type `Int`.\n */\npublic class
IntRange(start: Int, endInclusive: Int) : IntProgression(start, endInclusive, 1), ClosedRange<Int> {\n    override val
start: Int get() = first\n    override val endInclusive: Int get() = last\n\n    override fun contains(value: Int): Boolean =
first <= value && value <= last\n\n    /**\n     * Checks whether the range is empty.\n     *\n     * The range is empty
if its start value is greater than the end value.\n     */\n    override fun isEmpty(): Boolean = first > last\n\n
    override fun equals(other: Any?): Boolean =\n        other is IntRange && (isEmpty() && other.isEmpty()) ||\n        first == other.first && last == other.last)\n\n
    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * first + last)\n\n    override fun toString(): String =
"$first..$last"\n\n    companion object {\n        /** An empty range of values of type Int. */\n        public val
EMPTY: IntRange = IntRange(1, 0)\n    }\n}\n\n/**\n * A range of values of type `Long`.\n */\npublic class
LongRange(start: Long, endInclusive: Long) : LongProgression(start, endInclusive, 1), ClosedRange<Long> {\n    override
val start: Long get() = first\n    override val endInclusive: Long get() = last\n\n    override fun
contains(value: Long): Boolean = first <= value && value <= last\n\n    /**\n     * Checks whether the range is
empty.\n     *\n     * The range is empty if its start value is greater than
the end value.\n     */\n    override fun isEmpty(): Boolean = first > last\n\n    override fun equals(other: Any?):
Boolean =\n        other is LongRange && (isEmpty() && other.isEmpty()) ||\n        first == other.first && last ==
other.last)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor
(last ushr 32))).toInt()\n\n    override fun toString(): String = "$first..$last"\n\n    companion object {\n        /** An
empty range of values of type Long. */\n        public val EMPTY: LongRange = LongRange(1, 0)\n    }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\nimport
kotlin.random.Random\n\ninternal object EmptyIterator : ListIterator<Nothing> {\n    override fun hasNext():
Boolean = false\n    override fun hasPrevious(): Boolean = false\n    override fun nextIndex(): Int = 0\n    override
fun previousIndex(): Int = -1\n    override fun next(): Nothing = throw NoSuchElementException()\n    override fun
previous(): Nothing = throw NoSuchElementException()\n}\n\ninternal object EmptyList : List<Nothing>,
Serializable, RandomAccess {\n    private const val serialVersionUID: Long = -7390468764508069838L\n\n    override fun equals(other: Any?): Boolean = other is List<*> && other.isEmpty()\n    override fun hashCode(): Int
= 1\n    override fun toString(): String = "[]"\n\n    override val size: Int get() = 0\n    override fun isEmpty():

```

```

Boolean = true\n    override fun contains(element: Nothing): Boolean = false\n    override fun containsAll(elements:
Collection<Nothing>): Boolean = elements.isEmpty()\n    override fun get(index:
Int): Nothing = throw IndexOutOfBoundsException("\nEmpty list doesn't contain element at index $index.\n")\n
override fun indexOf(element: Nothing): Int = -1\n    override fun lastIndexOf(element: Nothing): Int = -1\n\n
override fun iterator(): Iterator<Nothing> = EmptyIterator\n    override fun listIterator(): ListIterator<Nothing> =
EmptyIterator\n    override fun listIterator(index: Int): ListIterator<Nothing> {\n        if (index != 0) throw
IndexOutOfBoundsException("\nIndex: $index\n")\n        return EmptyIterator\n    }\n\n    override fun
subList(fromIndex: Int, toIndex: Int): List<Nothing> {\n        if (fromIndex == 0 && toIndex == 0) return this\n
throw IndexOutOfBoundsException("\nfromIndex: $fromIndex, toIndex: $toIndex\n")\n    }\n\n    private fun
readResolve(): Any = EmptyList\n}\n\ninternal fun <T> Array<out T>.asCollection(): Collection<T> =
ArrayAsCollection(this, isVarargs = false)\n\nprivate class ArrayAsCollection<T>(val values: Array<out T>, val
isVarargs:
Boolean) : Collection<T> {\n    override val size: Int get() = values.size\n    override fun isEmpty(): Boolean =
values.isEmpty()\n    override fun contains(element: T): Boolean = values.contains(element)\n    override fun
containsAll(elements: Collection<T>): Boolean = elements.all { contains(it) }\n    override fun iterator():
Iterator<T> = values.iterator()\n    // override hidden toArray implementation to prevent copying of values array\n
public fun toArray(): Array<out Any?> = values.copyOfToArrayOfAny(isVarargs)\n}\n\n/**\n * Returns an empty
read-only list. The returned list is serializable (JVM).\n * @sample
samples.collections.Collections.Lists.emptyReadOnlyList\n */\n\npublic fun <T> emptyList(): List<T> =
EmptyList\n\n/**\n * Returns a new read-only list of given elements. The returned list is serializable (JVM).\n *
@sample samples.collections.Collections.Lists.readOnlyList\n */\n\npublic fun <T> listOf(vararg elements: T):
List<T> = if (elements.size > 0) elements.asList()\n    else emptyList()\n\n/**\n * Returns an empty read-only list. The returned list is serializable (JVM).\n * @sample
samples.collections.Collections.Lists.emptyReadOnlyList\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>
listOf(): List<T> = emptyList()\n\n/**\n * Returns an empty new [MutableList].\n * @sample
samples.collections.Collections.Lists.emptyMutableList\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableListOf(): MutableList<T> =
ArrayList()\n\n/**\n * Returns an empty new [ArrayList].\n * @sample
samples.collections.Collections.Lists.emptyArrayList\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> arrayListOf(): ArrayList<T> =
ArrayList()\n\n/**\n * Returns a new [MutableList] with the given elements.\n * @sample
samples.collections.Collections.Lists.mutableList\n */\n\npublic fun <T> mutableListOf(vararg elements: T):
MutableList<T> =\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements,
isVarargs = true))\n\n/**\n * Returns a new [ArrayList] with the given elements.\n * @sample
samples.collections.Collections.Lists.arrayList\n */\n\npublic fun <T> arrayListOf(vararg elements: T): ArrayList<T>
=\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))\n\n/**\n * Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The
returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\n\npublic fun
<T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()\n\n/**\n * Returns a new read-only list only of those given elements, that are not null. The returned list is serializable
(JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\n\npublic fun <T : Any>
listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()\n\n/**\n * Creates
a new read-only list with the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n * It\n * The function [init] is called for each list element sequentially starting from the first one.\n * It
should return the value for a list element given its index.\n * @sample
samples.collections.Collections.Lists.readOnlyListFromInitializer\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> List(size: Int, init: (index: Int) -> T):
List<T> = MutableList(size, init)\n\n/**\n * Creates a new mutable list with the specified [size], where each element

```

is calculated by calling the specified `[init]` function. The function `[init]` is called for each list element sequentially starting from the first one. It should return the value for a list element given its index.

`@sample samples.collections.Collections.Lists.mutableListFromInitializer`

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline
fun <T> MutableList(size: Int, init: (index: Int) -> T): MutableList<T> {\n    val list = ArrayList<T>(size)\n    repeat(size) { index -> list.add(init(index)) }\n    return list\n}\n\n/**\n * Builds a new read-only [List] by
populating a [MutableList] using the given [builderAction]\n * and returning a read-only list with the same
elements.\n * The list passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it
outside of the function produces an unspecified behavior.\n * The returned list is serializable (JVM).\n *\n *
@sample samples.collections.Builders.Lists.buildListSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>().() -> Unit): List<E> {\n    contract {
callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildListInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
expect inline fun <E> buildListInternal(builderAction: MutableList<E>().() -> Unit): List<E>\n\n/**\n * Builds a
new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list
with the same elements.\n * The list passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * The returned list is
serializable (JVM).\n * [capacity] is used to hint the expected number of elements added in the
[builderAction].\n * @throws IllegalArgumentException if the given [capacity] is negative.\n *\n * @sample
samples.collections.Builders.Lists.buildListSampleWithCapacity\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildList(capacity: Int, @BuilderInference builderAction: MutableList<E>().() -> Unit): List<E> {\n
contract
{ callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return buildListInternal(capacity,
builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>().() -> Unit): List<E>\n\n/**\n * Returns an
[IntRange] of the valid indices for this collection.\n * @sample
samples.collections.Collections.Collections.indicesOfCollection\n *\npublic val Collection<*>.indices: IntRange\n
get() = 0..size - 1\n\n/**\n * Returns the index of the last item in the list or -1 if the list is empty.\n * @sample
samples.collections.Collections.Lists.lastIndexOfList\n *\npublic val <T> List<T>.lastIndex: Int\n    get() =
this.size - 1\n\n/**\n * Returns `true` if the collection is not empty.\n * @sample
samples.collections.Collections.Collections.collectionIsNotEmpty\n *\n@kotlin.internal.InlineOnly\npublic inline
fun <T> Collection<T>.isNotEmpty(): Boolean = !isEmpty()\n\n/**\n * Returns `true` if this nullable collection is either null or empty.\n * @sample
samples.collections.Collections.Collections.collectionOrNullEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>?.isNullOrEmpty():
Boolean {\n    contract {\n        returns(false) implies (this@isNullOrEmpty != null)\n    }\n    return this == null ||
this.isEmpty()\n}\n\n/**\n * Returns this Collection if it's not `null` and the empty list otherwise.\n * @sample
samples.collections.Collections.Collections.collectionOrNullEmpty\n *\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()\n\n/**\n * Returns this List if it's not `null` and
the empty list otherwise.\n * @sample samples.collections.Collections.Lists.listOrNull\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>?.orEmpty(): List<T> = this ?: emptyList()\n\n/**\n *
Returns this collection if it's not empty\n * or the
result of calling [defaultValue] function if the collection is empty.\n *\n * @sample
samples.collections.Collections.Collections.collectionIfEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifEmpty(defaultValue: () ->
```



```

samples.collections.Collections.Lists.binarySearchByKey\n *\npublic inline fun <T, K : Comparable<K>>
List<T>.binarySearchBy(\n key: K?,\n fromIndex: Int = 0,\n toIndex: Int = size,\n crossinline selector: (T) ->
K?\n): Int =\n binarySearch(fromIndex, toIndex) { compareValues(selector(it), key) }\n\n// do not introduce this
overload --- too rare\n\npublic fun <T, K> List<T>.binarySearchBy(key: K, comparator: Comparator<K>,\n
fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =\n\n binarySearch(fromIndex, toIndex) {
comparator.compare(selector(it), key) }\n\n\n**\n * Searches this list or its range for an
element for which the given [comparison] function returns zero using the binary search algorithm.\n *\n * The list is
expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements,\n *
i.e. negative values come before zero and zeroes come before positive values.\n * Otherwise, the result is
undefined.\n *\n * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee
which one will be found.\n *\n * @param comparison function that returns zero when called on the list element
being searched.\n * On the elements coming before the target element, the function must return negative values;\n *
on the elements coming after the target element, the function must return positive values.\n *\n * @return the index
of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion
point `(-insertion point - 1)`.\n * The insertion point is defined as the index
at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n
*\n * @sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n *\npublic fun <T>
List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {\n rangeCheck(size,
fromIndex, toIndex)\n\n var low = fromIndex\n var high = toIndex - 1\n\n while (low <= high) {\n val mid
= (low + high).ushr(1) // safe from overflows\n val midVal = get(mid)\n val cmp = comparison(midVal)\n\n
if (cmp < 0)\n low = mid + 1\n else if (cmp > 0)\n high = mid - 1\n else\n return
mid // key found\n }\n\n return -(low + 1) // key not found\n}\n\n\n**\n * Checks that `from` and `to` are in\n * the
range of [0..size] and throws an appropriate exception, if they aren't.\n *\nprivate fun rangeCheck(size: Int,
fromIndex: Int, toIndex: Int) {\n when {\n\n fromIndex > toIndex -> throw IllegalArgumentException("\nfromIndex ($fromIndex) is greater than toIndex
($toIndex).")\n\n fromIndex < 0 -> throw IndexOutOfBoundsException("\nfromIndex ($fromIndex) is less than
zero.")\n\n toIndex > size -> throw IndexOutOfBoundsException("\ntoIndex ($toIndex) is greater than size
($size).")\n\n }\n}\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal expect fun checkIndexOverflow(index:
Int): Int\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal expect fun checkCountOverflow(count: Int):
Int\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal fun throwIndexOverflow() { throw
ArithmeticException("\nIndex overflow has happened.") }\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal fun
throwCountOverflow() { throw ArithmeticException("\nCount overflow has happened.") }\n\n\n"/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can
be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("\nMapsKt")\n@file:OptIn(kotlin.experimental.
ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\nprivate object
EmptyMap : Map<Any?, Nothing>, Serializable {\n private const val serialVersionUID: Long =
8246714829545688274\n\n override fun equals(other: Any?): Boolean = other is Map<*, *> &&
other.isEmpty()\n\n override fun hashCode(): Int = 0\n\n override fun toString(): String = "{}"\n\n override val
size: Int get() = 0\n\n override fun isEmpty(): Boolean = true\n\n\n override fun containsKey(key: Any?): Boolean =
false\n\n override fun containsValue(value: Nothing): Boolean = false\n\n\n override fun get(key: Any?): Nothing? =
null\n\n override val entries: Set<Map.Entry<Any?, Nothing>> get() = EmptySet\n\n override val keys: Set<Any?>
get() = EmptySet\n\n override val values: Collection<Nothing> get() = EmptyList\n\n\n private fun readResolve(): Any = EmptyMap\n}\n\n\n**\n * Returns an empty read-only map of specified type.\n
*\n * The returned map is serializable (JVM).\n *\n @sample
samples.collections.Maps.Instantiation.emptyReadOnlyMap\n *\npublic fun <K, V> emptyMap(): Map<K, V> =
@Suppress("\nUNCHECKED_CAST") (EmptyMap as Map<K, V>)\n\n\n**\n * Returns a new read-only map with

```

the specified contents, given as a list of pairs where the first value is the key and the second is the value. If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs. Entries of the map are iterated in the order they were specified. The returned map is serializable (JVM).

```

@sample samples.collections.Maps.Instantiation.mapFromPairs
public fun <K, V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> =
    if (pairs.size > 0) pairs.toMap(LinkedHashMap(mapCapacity(pairs.size))) else
    emptyMap()
Returns an empty read-only map.
The returned map is serializable (JVM).
@sample
samples.collections.Maps.Instantiation.emptyReadOnlyMap
public inline fun
<K, V> mapOf(): Map<K, V> = emptyMap()
Returns an empty new [MutableMap].
The returned
map preserves the entry iteration order.
@sample samples.collections.Maps.Instantiation.emptyMutableMap
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <K, V> mutableMapOf():
MutableMap<K, V> = LinkedHashMap()
Returns a new [MutableMap] with the specified contents, given
as a list of pairs where the first component is the key and the second is the value.
If multiple pairs have
the same key, the resulting map will contain the value from the last of those pairs.
Entries of the map are
iterated in the order they were specified.
@sample
samples.collections.Maps.Instantiation.mutableMapFromPairs
@sample
samples.collections.Maps.Instantiation.emptyMutableMap
public fun <K, V> mutableMapOf(vararg pairs: Pair<K, V>): MutableMap<K, V> =
    LinkedHashMap<K,
V>(mapCapacity(pairs.size)).apply { putAll(pairs) }
Returns an empty new [HashMap].
@sample
samples.collections.Maps.Instantiation.emptyHashMap
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <K, V> hashMapOf(): HashMap<K, V>
= HashMap<K, V>()
Returns a new [HashMap] with the specified contents, given as a list of pairs
where the first component is the key and the second is the value.
@sample
samples.collections.Maps.Instantiation.hashMapFromPairs
public fun <K, V> hashMapOf(vararg pairs:
Pair<K, V>): HashMap<K, V> = HashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }
Returns an empty new [LinkedHashMap].
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline
fun <K, V> linkedMapOf(): LinkedHashMap<K, V> = LinkedHashMap<K, V>()
Returns a new
[LinkedHashMap]
with the specified contents, given as a list of pairs where the first component is the key and the second is the
value.
If multiple pairs have the same key, the resulting map will contain the value from the last of those
pairs.
Entries of the map are iterated in the order they were specified.
@sample
samples.collections.Maps.Instantiation.linkedMapFromPairs
public fun <K, V> linkedMapOf(vararg pairs:
Pair<K, V>): LinkedHashMap<K, V> = pairs.toMap(LinkedHashMap(mapCapacity(pairs.size)))
Builds
a new read-only [Map] by populating a [MutableMap] using the given [builderAction] and returning a read-only
map with the same key-value pairs.
The map passed as a receiver to the [builderAction] is valid only inside
that function.
Using it outside of the function produces an unspecified behavior.
Entries of the map are
iterated in the order they were added by the [builderAction].
The returned map is serializable
(JVM).
@sample samples.collections.Builders.Maps.buildMapSample
@SinceKotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <K, V> buildMap(@BuilderInference builderAction: MutableMap<K, V>().->Unit): Map<K, V> {
    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }
    return
    buildMapInternal(builderAction)
}
@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>().->Unit): Map<K,
V>
Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction] and
returning a read-only map with the same key-value pairs.
The map passed as a receiver to the
[builderAction] is valid only inside that function.
Using it outside of the function produces an unspecified
behavior.
[capacity] is used to hint the expected number of pairs added in the

```

```

[builderAction].\n *\n * Entries of the map are iterated in the order they were added by the [builderAction].\n *\n *
The returned map is serializable (JVM).\n *\n * @throws IllegalArgumentException if the given [capacity] is
negative.\n *\n * @sample samples.collections.Builders.Maps.buildMapSample\n
*\n*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun <K, V> buildMap(capacity: Int, @BuilderInference builderAction: MutableMap<K, V>().-> Unit):
Map<K, V> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildMapInternal(capacity,
builderAction)\n}\n*\n*\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>().-> Unit): Map<K, V>\n\n/**\n *
Calculate the initial capacity of a map.\n *\n*\n@PublishedApi\ninternal expect fun mapCapacity(expectedSize: Int):
Int\n\n/**\n * Returns
`true` if this map is not empty.\n *\n * @sample samples.collections.Maps.Usage.mapIsNotEmpty\n
*\n*\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.isEmpty(): Boolean =
!isEmpty()\n\n/**\n * Returns `true` if this nullable map is either null or empty.\n *\n * @sample
samples.collections.Maps.Usage.mapIsNullOrEmpty\n
*\n*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>?.isNullOrEmpty(): Boolean {\n    contract {\n        returns(false) implies (this@isNullOrEmpty != null)\n    }\n\n    return this == null || isEmpty()\n}\n\n/**\n * Returns the [Map] if its not `null`, or the empty [Map] otherwise.\n
*\n * @sample samples.collections.Maps.Usage.mapOrEmpty\n
*\n*\n@kotlin.internal.InlineOnly\npublic inline fun
<K, V> Map<K, V>?.orEmpty(): Map<K, V> = this ?: emptyMap()\n\n/**\n * Returns this map if it's not empty\n *
or the result of calling [defaultValue] function if the map is empty.\n *\n * @sample
samples.collections.Maps.Usage.mapIfEmpty\n
*\n*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <M, R> M.ifEmpty(defaultValue: () ->
R): R where M : Map<*, *>, M : R =\n    if (isEmpty()) defaultValue() else this\n\n/**\n * Checks if the map
contains the given key.\n *\n * This method allows to use the `x` in map` syntax for checking whether an object is
contained in the map.\n *\n * @sample samples.collections.Maps.Usage.containsKey\n
*\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K,
V>.contains(key: K): Boolean = containsKey(key)\n\n/**\n * Returns the value corresponding to the given [key], or
`null` if such a key is not present in the map.\n *\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun
<@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.get(key: K): V? =\n    @Suppress("UNCHECKED_CAST") (this as Map<K, V>).get(key)\n\n/**\n * Allows to use the index operator
for storing values in a mutable map.\n *\n*\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<K, V>.set(key: K, value: V): Unit {\n    put(key, value)\n}\n\n/**\n *
Returns `true` if the map contains the specified [key].\n *\n * Allows to overcome type-safety restriction of
`containsKey` that requires to pass a key of type `K`.\n *\n*\n@kotlin.internal.InlineOnly\npublic inline fun
<@kotlin.internal.OnlyInputTypes K> Map<out K, *>.containsKey(key: K): Boolean =\n    @Suppress("UNCHECKED_CAST") (this as Map<K, *>).containsKey(key)\n\n/**\n * Returns `true` if the map
maps one or more keys to the specified [value].\n *\n * Allows to overcome type-safety restriction of
`containsValue` that requires to pass a value of type `V`.\n *\n * @sample
samples.collections.Maps.Usage.containsValue\n
*\n*\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")
// false warning, extension takes precedence in some cases\n@kotlin.internal.InlineOnly\npublic inline fun <K,
@kotlin.internal.OnlyInputTypes V> Map<K, V>.containsValue(value: V): Boolean
= this.containsValue(value)\n\n/**\n * Removes the specified key and its corresponding value from this map.\n
*\n * @return the previous value associated with the key, or `null` if the key was not present in the map.\n\n *
Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.\n
*\n*\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> MutableMap<out K,
V>.remove(key: K): V? =\n    @Suppress("UNCHECKED_CAST") (this as MutableMap<K,
V>).remove(key)\n\n/**\n * Returns the key component of the map entry.\n *\n * This method allows to use

```


destructuring declarations when working with maps, for example:

```

for ((key, value) in map) {
    // do something with the key and the value
}

```

This method allows to use destructuring declarations when working with maps, for example:

```

for ((key, value) in map) {
    // do something with the key and the value
}

```

`component1(): K = key` Returns the value component of the map entry.

`component2(): V = value` Converts entry to [Pair] with key being first component and value being second.

`toPair(): Pair<K, V> = Pair(key, value)` Returns the value for the given key, or the result of the [defaultValue] function if there was no entry for the given key.

`getOrElse(): V` Returns the value for the given [key] or throws an exception if there is no such key in the map. If the map was created by [withDefault], resorts to its `defaultValue` provider function instead of throwing an exception. @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default value was provided for that map.

`getValue(key: K): V` Returns the value for the given key. If the key is not found in the map, calls the [defaultValue] function, puts its result into the map under the given key and returns it. Note that the operation is not guaranteed to be atomic if the map is being modified concurrently.

`getOrElsePut(key: K, default: () -> V): V` Returns an [Iterator] over the entries in the [Map].

`forOverEntries(): Iterator<Map.Entry<K, V>>` Returns a [MutableIterator] over the mutable entries in the [MutableMap].

`iterator(): MutableIterator<MutableMap.MutableEntry<K, V>>` Returns an [Iterator] over the entries in the [Map].

`mapValuesTo(destination: M, transform: (Map.Entry<K, V>) -> R): M` Populates the given [destination] map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map. In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

`mapKeysTo(destination: M, transform: (Map.Entry<K, V>) -> R): M` Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.

`putAll(pairs: Array<Pair<K, V>>): Unit` Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.

`putAll(pairs: Iterable<Pair<K, V>>): Unit` Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.

`putAll(pairs: Sequence<Pair<K, V>>): Unit` Returns a new map with entries having the keys of this map and the values obtained by applying the [transform] function to each entry in this [Map]. The returned map preserves the entry iteration order of

```

the original map.\n *\n * @sample samples.collections.Maps.Transformations.mapValues\n *\npublic
inline fun <K, V, R> Map<out K, V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {\n return
mapValuesTo(LinkedHashMap<K, R>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n\n/**\n *
Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this\n
* [Map] and the values of this map.\n *\n * In case if any two entries are mapped to the equal keys, the value of the
latter one will overwrite\n * the value associated with the former one.\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n * @sample samples.collections.Maps.Transformations.mapKeys\n
*\npublic inline fun <K, V, R> Map<out K, V>.mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {\n
return mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n\n/**\n
* Returns a map containing all key-value pairs with keys matching the given [predicate].\n *\n * The returned map
preserves
preserves
the entry iteration order of the original map.\n * @sample samples.collections.Maps.Filtering.filterKeys\n
*\npublic inline fun <K, V> Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {\n val result =
LinkedHashMap<K, V>()\n for (entry in this) {\n if (predicate(entry.key)) {\n result.put(entry.key,
entry.value)\n } }\n return result}\n\n/**\n * Returns a map containing all key-value pairs with values
matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original map.\n
*\n * @sample samples.collections.Maps.Filtering.filterValues\n *\npublic inline fun <K, V> Map<out K,
V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n for (entry
in this) {\n if (predicate(entry.value)) {\n result.put(entry.key, entry.value)\n } }\n return
result}\n\n/**\n * Appends all entries matching the given [predicate] into
the mutable map given as [destination] parameter.\n *\n * @return the destination map.\n * @sample
samples.collections.Maps.Filtering.filterTo\n *\npublic inline fun <K, V, M : MutableMap<in K, in V>> Map<out
K, V>.filterTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {\n for (element in this) {\n if
(predicate(element)) {\n destination.put(element.key, element.value)\n } }\n return
destination}\n\n/**\n * Returns a new map containing all key-value pairs matching the given [predicate].\n *\n *
The returned map preserves the entry iteration order of the original map.\n *\n * @sample
samples.collections.Maps.Filtering.filter\n *\npublic inline fun <K, V> Map<out K, V>.filter(predicate:
(Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return filterTo(LinkedHashMap<K, V>(),
predicate)\n}\n\n/**\n * Appends all entries not matching the given [predicate] into the given [destination].\n
*\n * @return the destination map.\n * @sample samples.collections.Maps.Filtering.filterNotTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterNotTo(destination: M, predicate:
(Map.Entry<K, V>) -> Boolean): M {\n for (element in this) {\n if (!predicate(element)) {\n
destination.put(element.key, element.value)\n } }\n return destination}\n\n/**\n * Returns a new map
containing all key-value pairs not matching the given [predicate].\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n * @sample samples.collections.Maps.Filtering.filterNot\n *\npublic inline fun
<K, V> Map<out K, V>.filterNot(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return
filterNotTo(LinkedHashMap<K, V>(), predicate)\n}\n\n/**\n * Returns a new map containing all key-value pairs
from the given collection of pairs.\n *\n * The returned map preserves the entry iteration order of the original
collection.\n *\n * If any of two pairs would have the same key the last
one gets added to the map.\n *\npublic fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {\n if (this is
Collection) {\n return when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(if (this is List) this[0] else
iterator().next())\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n } }\n return
toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n}\n\n/**\n * Populates and returns the [destination]
mutable map with key-value pairs from the given collection of pairs.\n *\npublic fun <K, V, M : MutableMap<in K,
in V>> Iterable<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n *
Returns a new map containing all key-value pairs from the given array of pairs.\n *\n * The returned map preserves
the entry iteration order of the original array.\n *\n * If any of two pairs would have the same key the last one gets added
to the map.\n *\npublic fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> =

```

```

when (size) {n 0 -> emptyMap()\n 1 -> mapOf(this[0])\n else -> toMap(LinkedHashMap<K,
V>(mapCapacity(size)))\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs
from the given array of pairs.\n *\npublic fun <K, V, M : MutableMap<in K, in V>> Array<out Pair<K,
V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map
containing all key-value pairs from the given sequence of pairs.\n *\n * The returned map preserves the entry
iteration order of the original sequence.\n * If any of two pairs would have the same key the last one gets added to
the map.\n *\npublic fun <K, V, M : MutableMap<in K, in V>> Sequence<Pair<K,
V>>.toMap(destination: M): M =\n destination.apply
{ putAll(this@toMap) }\n\n/**\n * Returns a new read-only map containing all key-value pairs from the original
map.\n *\n * The returned map preserves the entry iteration order of the original map.\n
*\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {n 0 ->
emptyMap()\n 1 -> toSingletonMap()\n else -> toMutableMap()\n}\n\n/**\n * Returns a new mutable map
containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration order of
the original map.\n *\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMutableMap():
MutableMap<K, V> = LinkedHashMap(this)\n\n/**\n * Populates and returns the [destination] mutable map with
key-value pairs from the given map.\n *\n@SinceKotlin("1.1")\npublic fun <K, V, M : MutableMap<in K, in V>>
Map<out K, V>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Creates a new
read-only map by replacing or adding an
entry to this map from a given key-value [pair].\n *\n * The returned map preserves the entry iteration order of the
original map.\n * The [pair] is iterated in the end if it has a unique key.\n *\npublic operator fun <K, V> Map<out
K, V>.plus(pair: Pair<K, V>): Map<K, V> =\n if (this.isEmpty()) mapOf(pair) else LinkedHashMap(this).apply {
put(pair.first, pair.second) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from
a given collection of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original
map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.\n *\npublic
operator fun <K, V> Map<out K, V>.plus(pairs: Iterable<Pair<K, V>>): Map<K, V> =\n if (this.isEmpty())
pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by
replacing or adding entries to this map from a given array of key-value [pairs].\n *\n * The
returned map preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated
in the end in the order of [pairs] array.\n *\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Array<out
Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs)
}\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given sequence of key-
value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those [pairs] with
unique keys are iterated in the end in the order of [pairs] sequence.\n *\npublic operator fun <K, V> Map<out K,
V>.plus(pairs: Sequence<Pair<K, V>>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(pairs)
}.optimizeReadOnlyMap()\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from
another [map].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those
entries of another [map] that are missing in this map are iterated in the end in the order of that [map].\n *\npublic
operator fun <K, V> Map<out K, V>.plus(map: Map<out K, V>): Map<K, V> =\n LinkedHashMap(this).apply {
putAll(map) }\n\n/**\n * Appends or replaces the given [pair] in this mutable map.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pair:
Pair<K, V>) {\n put(pair.first, pair.second)\n}\n\n/**\n * Appends or replaces all pairs from the given collection
of [pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V>
MutableMap<in K, in V>.plusAssign(pairs: Iterable<Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or
replaces all pairs from the given array of [pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Array<out Pair<K, V>>) {\n
putAll(pairs)\n}\n\n/**\n * Appends or

```

```

replaces all pairs from the given sequence of [pairs] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Sequence<Pair<K, V>>) {\n
putAll(pairs)\n}\n\n/**\n * Appends or replaces all entries from the given [map] in this mutable map.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(map:
Map<K, V>) {\n  putAll(map)\n}\n\n/**\n * Returns a map containing all entries of the original map except the
entry with the given [key].\n * The returned map preserves the entry iteration order of the original map.\n
*/\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K, V>.minus(key: K): Map<K, V> =\n
this.toMutableMap().apply { minusAssign(key) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all
entries of the original map except those entries\n * the keys of which are contained in the given [keys] collection.\n
*/\n * The returned map preserves
the entry iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(keys: Iterable<K>): Map<K, V> =\n  this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] array.\n * The returned map preserves the entry iteration
order of the original map.\n */\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K, V>.minus(keys:
Array<out K>): Map<K, V> =\n  this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] sequence.\n * The returned map preserves the entry
iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(keys: Sequence<K>): Map<K, V>
=\n  this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()\n\n/**\n * Removes the entry
with the given [key] from this mutable map.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<K, V>.minusAssign(key: K) {\n  remove(key)\n}\n\n/**\n * Removes all
entries the keys of which are contained in the given [keys] collection from this mutable map.\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Iterable<K>) {\n  this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries the keys of
which are contained in the given [keys] array from this mutable map.\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Array<out K>) {\n  this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries from the
keys of which are contained in the given [keys] sequence from this mutable map.\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Sequence<K>) {\n  this.keys.removeAll(keys)\n}\n\n// do not expose for now
@PublishedApi\ninternal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) {\n  0 -> emptyMap()\n
1 -> toSingletonMapOrSelf()\n  else -> this\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n@file:OptIn(kotlin.experimenta
l.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\ninternal object
EmptySet : Set<Nothing>, Serializable {\n  private const val serialVersionUID: Long =
3406603774387020532\n\n  override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()\n
  override fun hashCode(): Int = 0\n  override fun toString(): String = "[]"\n\n  override val size: Int get() = 0\n
  override fun isEmpty(): Boolean = true\n  override fun contains(element: Nothing): Boolean = false\n  override
fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()\n\n  override fun iterator():
Iterator<Nothing> = EmptyIterator\n\n  private fun readResolve(): Any = EmptySet\n}\n\n\n/**\n * Returns an
empty read-only set. The returned set is serializable (JVM).\n * @sample
samples.collections.Collections.Sets.emptyReadOnlySet\n */\n\npublic fun <T> emptySet(): Set<T> =
EmptySet\n\n/**\n * Returns a new read-only set with the given elements.\n * Elements of the set are iterated in the
order they were specified.\n * The returned set is serializable (JVM).\n * @sample

```

```

samples.collections.Collections.Sets.readOnlySet\n * \npublic fun <T> setOf(vararg elements: T): Set<T> = if
(elements.size > 0) elements.toSet() else emptySet()\n\n/**\n
 * Returns an empty read-only set. The returned set is serializable (JVM).\n * @sample
samples.collections.Collections.Sets.emptyReadOnlySet\n * \n@kotlin.internal.InlineOnly\npublic inline fun <T>
setOf(): Set<T> = emptySet()\n\n/**\n * Returns an empty new [MutableSet].\n * \n * The returned set preserves the
element iteration order.\n * @sample samples.collections.Collections.Sets.emptyMutableSet\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableSetOf(): MutableSet<T> =
LinkedHashSet()\n\n/**\n * Returns a new [MutableSet] with the given elements.\n * Elements of the set are
iterated in the order they were specified.\n * @sample samples.collections.Collections.Sets.mutableSet\n * \npublic
fun <T> mutableSetOf(vararg elements: T): MutableSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/** Returns an empty new [HashSet].
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> hashSetOf(): HashSet<T>
= HashSet()\n\n/** Returns a new [HashSet] with the given elements. *\npublic fun <T> hashSetOf(vararg
elements: T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))\n\n/** Returns an
empty new [LinkedHashSet].\n * @sample samples.collections.Collections.Sets.emptyLinkedHashSet\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> linkedSetOf(): LinkedHashSet<T>
= LinkedHashSet()\n\n/**\n * Returns a new [LinkedHashSet] with the given elements.\n * Elements of the set are
iterated in the order they were specified.\n * @sample samples.collections.Collections.Sets.linkedHashSet\n
*\npublic fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/**\n * Returns a new read-only set either
with single given element, if it is not null, or empty set if the element is null.\n * The returned set is serializable
(JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n
*\n@SinceKotlin("1.4")\npublic fun <T : Any> setOfNotNull(element: T?): Set<T> = if (element != null)
setOf(element) else emptySet()\n\n/**\n * Returns a new read-only set only with those given elements, that are not
null.\n * Elements of the set are iterated in the order they were specified.\n * The returned set is serializable
(JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n *\n@SinceKotlin("1.4")\npublic fun <T
: Any> setOfNotNull(vararg elements: T?): Set<T> {\n    return
elements.filterNotNullTo(LinkedHashSet())\n}\n\n/**\n * Builds a new read-only [Set] by populating a
[MutableSet] using the given [builderAction]\n * and returning a read-only set with the same elements.\n * \n * The
set passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function
produces an unspecified behavior.\n * \n * Elements of the set are iterated in the order they were added by the
[builderAction].\n * \n * The
returned set is serializable (JVM).\n * \n * @sample samples.collections.Builders.Sets.buildSetSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    contract {
callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildSetInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
expect inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E>\n\n/**\n * Builds a
new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a read-only set
with the same elements.\n * \n * The set passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * [capacity] is used to hint the
expected number of elements added in the
[builderAction].\n * \n * Elements of the set are iterated in the order they were added by the [builderAction].\n * \n *
The returned set is serializable (JVM).\n * \n * @throws IllegalArgumentException if the given [capacity] is
negative.\n * \n * @sample samples.collections.Builders.Sets.buildSetSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {\n
contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return buildSetInternal(capacity,

```

```

builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit): Set<E>\n\n/** Returns this Set
if it's not `null` and the empty set otherwise. */\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Set<T>?.orEmpty(): Set<T> = this
?: emptySet()\n\ninternal fun <T> Set<T>.optimizeReadOnlySet() = when (size) {\n  0 -> emptySet()\n  1 ->
setOf(iterator().next())\n  else -> this\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n@file:Suppress("PLATFOR
M_CLASS_MAPPED_TO_KOTLIN")\n\npackage kotlin.text\n\n/**\n * Parses the string as a signed [Byte]
number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toByteArrayOrNull(): Byte? = toByteOrNull(radix = 10)\n\n/**\n * Parses
the string as a signed [Byte] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n * \n * @throws IllegalArgumentException when [radix] is not a valid radix for
string to number conversion.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toByteArrayOrNull(radix: Int): Byte? {\n  val int = this.toIntOrNull(radix) ?: return null\n  if (int < Byte.MIN_VALUE || int > Byte.MAX_VALUE) return
null\n  return int.toByteArray()\n}\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if
the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toShortOrNull(): Short? = toShortOrNull(radix = 10)\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or
`null` if the string is not a valid representation of a number.\n * \n * @throws IllegalArgumentException when
[radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.1")\n\npublic fun
String.toShortOrNull(radix: Int): Short? {\n  val int = this.toIntOrNull(radix) ?: return null\n  if (int <
Short.MIN_VALUE || int > Short.MAX_VALUE) return null\n  return int.toShort()\n}\n\n/**\n *
Parses the string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toIntOrNull(): Int? = toIntOrNull(radix = 10)\n\n/**\n *
Parses the string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n * \n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number
conversion.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toIntOrNull(radix: Int): Int? {\n  checkRadix(radix)\n\n  val length = this.length\n  if (length == 0) return null\n\n  val start: Int\n  val isNegative: Boolean\n  val limit:
Int\n\n  val firstChar = this[0]\n  if (firstChar < '0') { // Possible leading sign\n    if (length == 1) return null //
non-digit (possible sign) only, no digits after\n\n    start = 1\n    if (firstChar == '-') {\n      isNegative =
true\n      limit = Int.MIN_VALUE\n    } else if (firstChar == '+') {\n      isNegative = false\n      limit = -Int.MAX_VALUE\n    } else\n    return null\n  } else {\n    start = 0\n    isNegative = false\n    limit = -Int.MAX_VALUE\n  }\n\n  val
limitForMaxRadix = (-Int.MAX_VALUE) / 36\n  var limitBeforeMul = limitForMaxRadix\n  var result = 0\n  for (i in start until length) {\n    val digit = digitOf(this[i], radix)\n\n    if (digit < 0) return null\n    if (result <
limitBeforeMul) {\n      if (limitBeforeMul == limitForMaxRadix) {\n        limitBeforeMul = limit /
radix\n\n        if (result < limitBeforeMul) {\n          return null\n        }\n      } else {\n        return null\n      }\n    }\n\n    result *= radix\n\n    if (result < limit + digit) return null\n\n    result -=
digit\n  }\n\n  return if (isNegative) result else -result\n}\n\n/**\n * Parses
the string as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toLongOrNull(): Long? = toLongOrNull(radix =
10)\n\n/**\n * Parses the string as a [Long] number and returns the result\n * or `null` if the string is not a valid
representation of a number.\n * \n * @throws IllegalArgumentException when [radix] is not a valid radix for string
to number conversion.\n
*\n@SinceKotlin("1.1")\n\npublic fun String.toLongOrNull(radix: Int): Long? {\n  checkRadix(radix)\n\n  val length = this.length\n  if (length == 0) return null\n\n  val start: Int\n  val isNegative:
Boolean\n  val limit: Long\n\n  val firstChar = this[0]\n  if (firstChar < '0') { // Possible leading sign\n    if
(length == 1) return null // non-digit (possible sign) only, no digits after\n\n    start = 1\n    if (firstChar == '-')

```

```

{\n      isNegative = true\n      limit = Long.MIN_VALUE\n
    } else if (firstChar == '+') {\n      isNegative = false\n      limit = -Long.MAX_VALUE\n    } else\n
    return null\n  } else {\n    start = 0\n    isNegative = false\n    limit = -Long.MAX_VALUE\n  }\n\nval limitForMaxRadix = (-Long.MAX_VALUE) / 36\n  var limitBeforeMul = limitForMaxRadix\n  var result = 0L\n  for (i in start until length) {\n    val digit = digitOf(this[i], radix)\n    if (digit < 0) return null\n    if (result < limitBeforeMul) {\n      if (limitBeforeMul == limitForMaxRadix) {\n        limitBeforeMul = limit / radix\n      }\n      if (result < limitBeforeMul) {\n        return null\n      }\n    } else {\n      return null\n    }\n  }\n  result *= radix\n  if (result < limit + digit) return null\n  result -= digit\n}\n\nreturn if (isNegative) result else -result\n}\n\ninternal\nfun numberFormatError(input: String): Nothing = throw NumberFormatException("Invalid number format: '$input')\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\nimport kotlin.contracts.*\nimport kotlin.jvm.JvmInline\nimport kotlin.math.*\n\n/**\n * Represents the amount of time one instant of time is away from another instant.\n * A negative duration is possible in a situation when the second instant is earlier than the first one.\n * The type can store duration values up to 1146 years with nanosecond precision,\n * and up to 1146 million years with millisecond precision.\n * If a duration-returning operation provided in `kotlin.time` produces a duration value that doesn't fit into the above range,\n * the returned `Duration` is infinite.\n * An infinite duration value [Duration.INFINITE] can be used to represent infinite timeouts.\n * To construct a duration use either the extension function [toDuration],\n * or the extension properties [hours], [minutes], [seconds], and so on,\n * available on [Int], [Long], and [Double] numeric types.\n * To get the value of this duration expressed in a particular [duration units][DurationUnit]\n * use the functions [toInt], [toLong], and [toDouble]\n * or the properties [inWholeHours], [inWholeMinutes], [inWholeSeconds], [inWholeNanoseconds], and so on.\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@JvmInline\npublic value class Duration\n\ninternal constructor(private val rawValue: Long) : Comparable<Duration> {\n  private val value: Long\n  get() = rawValue shr 1\n  private inline val unitDiscriminator: Int\n  get() = rawValue.toInt() and 1\n  private fun isInNanos() = unitDiscriminator == 0\n  private fun isInMillis() = unitDiscriminator == 1\n  private val storageUnit\n  get()\n\n  = if (isInNanos()) DurationUnit.NANOSECONDS else DurationUnit.MILLISECONDS\n\n  init {\n    if (durationAssertionsEnabled) {\n      if (isInNanos()) {\n        if (value !in -MAX_NANOS..MAX_NANOS)\n          throw AssertionError("$value ns is out of nanoseconds range")\n      } else {\n        if (value !in -MAX_MILLIS..MAX_MILLIS)\n          throw AssertionError("$value ms is out of milliseconds range")\n        if (value in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS)\n          throw AssertionError("$value ms is denormalized")\n      }\n    }\n  }\n\n  companion object {\n    /** The duration equal to exactly 0 seconds. */\n    public val ZERO: Duration = Duration(0L)\n    /** The duration whose value is positive infinity. It is useful for representing timeouts that should never expire. */\n    public val INFINITE: Duration = durationOfMillis(-MAX_MILLIS)\n\n    internal val NEG_INFINITE: Duration = durationOfMillis(-MAX_MILLIS)\n\n    /** Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified [targetUnit]. */\n    @ExperimentalTime\n    public fun convert(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double =\n      convertDurationUnit(value, sourceUnit, targetUnit)\n\n    // Duration construction extension properties in Duration companion scope\n    /** Returns a [Duration] equal to this [Int] number of nanoseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.nanoseconds\n    get() = toDuration(DurationUnit.NANOSECONDS)\n    /** Returns a [Duration] equal to this [Long] number of nanoseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.nanoseconds\n    get() = toDuration(DurationUnit.NANOSECONDS)\n    /** Returns a [Duration] equal to this [Double] number of nanoseconds. */\n    * Depending on its magnitude, the value is rounded to an integer

```

```

number of nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException if this [Double]
value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)\n\n    /** Returns a [Duration] equal to this [Int] number of
microseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n    /** Returns a [Duration] equal to this [Long] number of
microseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n    /**\n    * Returns a [Duration] equal to this [Double]
number of microseconds.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException if this [Double] value
is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n    /** Returns a [Duration] equal to this [Int] number of
milliseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] equal to this [Long] number of
milliseconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n    /**\n    * Returns a [Duration] equal to this [Double]
number of milliseconds.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number of
nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n\n    /** Returns a [Duration] equal to this [Int] number of seconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n    /** Returns a [Duration] equal to
this [Long] number of seconds. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n    /**\n    * Returns a [Duration] equal to this [Double] number of
seconds.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or
milliseconds.\n    *\n    * @throws IllegalArgumentException if this [Double] value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n\n    /** Returns a [Duration] equal to this [Int] number of minutes. */\n    @kotlin.internal.InlineOnly\n    public inline
val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] equal to this [Long]
number of minutes. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.minutes get() =
toDuration(DurationUnit.MINUTES)\n\n    /**\n    * Returns a [Duration] equal to this [Double] number of
minutes.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or
milliseconds.\n    *\n    * @throws IllegalArgumentException if this [Double] value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.minutes get() =
toDuration(DurationUnit.MINUTES)\n\n\n    /** Returns a [Duration] equal to this [Int] number of hours. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n    /**
Returns a [Duration] equal to this [Long] number of hours. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n    /**\n    * Returns a
[Duration] equal to this [Double] number of hours.\n    *\n    * Depending on its magnitude, the value is
rounded to an integer number of nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException
if this [Double] value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.hours
get() = toDuration(DurationUnit.HOURS)\n\n\n    /** Returns a [Duration] equal to this [Int] number of days. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.days get() = toDuration(DurationUnit.DAYS)\n\n    /**
Returns a [Duration] equal to this [Long] number of days. */\n    @kotlin.internal.InlineOnly\n    public inline
val Long.days get() = toDuration(DurationUnit.DAYS)\n\n    /**\n    * Returns a [Duration] equal to this
[Double] number of days.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException
if this [Double] value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.days get() = toDuration(DurationUnit.DAYS)\n\n\n    /** Returns a [Duration] equal to this [Int] number of
years. */\n    @kotlin.internal.InlineOnly\n    public inline val Int.years get() = toDuration(DurationUnit.YEARS)\n\n    /** Returns a [Duration] equal to this [Long] number of
years. */\n    @kotlin.internal.InlineOnly\n    public inline val Long.years get() = toDuration(DurationUnit.YEARS)\n\n    /**\n    * Returns a [Duration] equal to this [Double] number of
years.\n    *\n    * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n    *\n    * @throws IllegalArgumentException
if this [Double] value is `NaN`.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline val Double.years get() = toDuration(DurationUnit.YEARS)

```



```

@kotlin.internal.InlineOnly\n    public inline val Double.days get() = toDuration(DurationUnit.DAYS)\n\n\n// deprecated static factory functions\n\n    /** Returns a [Duration] representing the specified [value] number of
nanoseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Int.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("value.nanoseconds",
"\kotlin.time.Duration.Companion.nanoseconds"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\npublic fun nanoseconds(value: Int): Duration = value.toDuration(DurationUnit.NANOSECONDS)\n\n    /**
Returns a [Duration] representing the specified [value] number of nanoseconds. */\n    @SinceKotlin("1.5")\n\n    @ExperimentalTime\n    @Deprecated("Use 'Long.nanoseconds' extension property from Duration.Companion
instead.", ReplaceWith("value.nanoseconds", "\kotlin.time.Duration.Companion.nanoseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun nanoseconds(value: Long): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n\n    /**\n    * Returns a [Duration] representing the
specified [value] number of nanoseconds.\n    *\n    * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n\n    @Deprecated("Use 'Double.nanoseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.nanoseconds", "\kotlin.time.Duration.Companion.nanoseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun nanoseconds(value: Double): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n\n    /**
Returns a [Duration] representing the specified [value] number of microseconds. */\n    @SinceKotlin("1.5")\n\n    @ExperimentalTime\n    @Deprecated("Use 'Int.microseconds' extension property from Duration.Companion
instead.", ReplaceWith("value.microseconds", "\kotlin.time.Duration.Companion.microseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun microseconds(value: Int): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n\n    /** Returns a [Duration] representing the specified
[value] number of microseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n\n    @Deprecated("Use 'Long.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "\kotlin.time.Duration.Companion.microseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun microseconds(value: Long): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n\n    /**\n    * Returns a [Duration] representing the specified [value] number of microseconds.\n    *\n    * @throws
IllegalArgumentException if the provided `Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n\n    @ExperimentalTime\n    @Deprecated("Use 'Double.microseconds' extension property from
Duration.Companion instead.", ReplaceWith("value.microseconds",
"\kotlin.time.Duration.Companion.microseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun microseconds(value: Double): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n\n    /** Returns a [Duration] representing the specified [value] number of milliseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Int.milliseconds' extension property
from Duration.Companion instead.", ReplaceWith("value.milliseconds",
"\kotlin.time.Duration.Companion.milliseconds"))\n\n    @DeprecatedSinceKotlin(warningSince
= "1.6")\n\n    public fun milliseconds(value: Int): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n\n    /** Returns a [Duration] representing the specified
[value] number of milliseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n\n    @Deprecated("Use 'Long.milliseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.milliseconds", "\kotlin.time.Duration.Companion.milliseconds"))\n\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n\n    public fun milliseconds(value: Long): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n\n    /**\n    * Returns a [Duration] representing the
specified [value] number of milliseconds.\n    *\n    * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n\n    @Deprecated("Use 'Double.milliseconds' extension property from Duration.Companion instead.",

```

```

ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun milliseconds(value: Double): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of seconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Int.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"kotlin.time.Duration.Companion.seconds"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun seconds(value: Int): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a [Duration]
representing the specified [value] number of seconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Long.seconds' extension property from Duration.Companion instead.",
ReplaceWith("value.seconds", "kotlin.time.Duration.Companion.seconds"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun seconds(value: Long): Duration =
value.toDuration(DurationUnit.SECONDS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of seconds.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Double.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"kotlin.time.Duration.Companion.seconds"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun seconds(value: Double): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a
[Duration] representing the specified [value] number of minutes. */\n    @SinceKotlin("1.5")\n
    @ExperimentalTime\n    @Deprecated("Use 'Int.minutes' extension property
from Duration.Companion instead.", ReplaceWith("value.minutes",
"kotlin.time.Duration.Companion.minutes"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun minutes(value: Int): Duration = value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration]
representing the specified [value] number of minutes. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Long.minutes' extension property from Duration.Companion instead.",
ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Long): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of minutes.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Double.minutes' extension property from Duration.Companion instead.",
ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Double): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] representing the specified
[value] number of hours. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Int.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"kotlin.time.Duration.Companion.hours"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Int): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of hours. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Long.hours' extension property from
Duration.Companion instead.", ReplaceWith("value.hours", "kotlin.time.Duration.Companion.hours"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun hours(value: Long): Duration =
value.toDuration(DurationUnit.HOURS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of hours.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Double.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"kotlin.time.Duration.Companion.hours"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Double): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of days. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n

```

```

@Deprecated("Use
'Int.days' extension property from Duration.Companion instead.", ReplaceWith("value.days",
"\"kotlin.time.Duration.Companion.days\""))\n    @DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public
fun days(value: Int): Duration = value.toDuration(DurationUnit.DAYS)\n    /** Returns a [Duration]
representing the specified [value] number of days. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Long.days' extension property from Duration.Companion instead.",
ReplaceWith("value.days", "\"kotlin.time.Duration.Companion.days\""))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun days(value: Long): Duration =
value.toDuration(DurationUnit.DAYS)\n    /**\n    * Returns a [Duration] representing the specified [value]
number of days.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n
*\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Double.days' extension property from Duration.Companion instead.",
ReplaceWith("value.days", "\"kotlin.time.Duration.Companion.days\""))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun days(value: Double): Duration =
value.toDuration(DurationUnit.DAYS)\n    /**\n    * Parses a string that represents a duration and returns the
parsed [Duration] value.\n    *\n    * The following formats are accepted:\n    *\n    * - ISO-8601
Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and [parseIsoString].\n    * - The format of string
returned by the default [Duration.toString] and `toString` in a specific unit,\n    * e.g. `10s`, `1h 30m` or `-(1h
30m)`.\n    *\n    * @throws IllegalArgumentException if the string doesn't represent a duration in any of the
supported formats.\n    * @sample samples.time.Durations.parse\n    */\n    public fun parse(value:
String): Duration = try {\n        parseDuration(value, strictIso = false)\n    } catch (e:
IllegalArgumentException) {\n        throw IllegalArgumentException("Invalid duration string format: '$value'.",
e)\n    }\n    /**\n    * Parses a string that represents a duration in ISO-8601 format and returns the parsed
[Duration] value.\n    *\n    * @throws IllegalArgumentException if the string doesn't represent a duration in
ISO-8601 format.\n    * @sample samples.time.Durations.parseIsoString\n    */\n    public fun
parseIsoString(value: String): Duration = try {\n        parseDuration(value, strictIso = true)\n    } catch (e:
IllegalArgumentException) {\n        throw IllegalArgumentException("Invalid ISO duration string format:
'$value'.", e)\n    }\n    /**\n    * Parses a string that represents a duration and returns the parsed [Duration]
value,\n    * or `null` if the string doesn't represent
a duration in any of the supported formats.\n    *\n    * The following formats are accepted:\n    *\n    * -
ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and [parseIsoString].\n    * - The format
of string returned by the default [Duration.toString] and `toString` in a specific unit,\n    * e.g. `10s`, `1h 30m`
or `-(1h 30m)`.\n    * @sample samples.time.Durations.parse\n    */\n    public fun parseOrNull(value:
String): Duration? = try {\n        parseDuration(value, strictIso = false)\n    } catch (e:
IllegalArgumentException) {\n        null\n    }\n    /**\n    * Parses a string that represents a duration in
ISO-8601 format and returns the parsed [Duration] value,\n    * or `null` if the string doesn't represent a duration
in ISO-8601 format.\n    * @sample samples.time.Durations.parseIsoString\n    */\n    public fun
parseIsoStringOrNull(value: String): Duration? =
try {\n        parseDuration(value, strictIso = true)\n    } catch (e: IllegalArgumentException) {\n        null\n
    }\n    }\n    // arithmetic operators\n    /** Returns the negative of this value. *\n    public operator fun
unaryMinus(): Duration = durationOf(-value, unitDiscriminator)\n    /**\n    * Returns a duration whose value is
the sum of this and [other] duration values.\n    *\n    * @throws IllegalArgumentException if the operation results
in an undefined value for the given arguments,\n    * e.g. when adding infinite durations of different sign.\n    */\n
public operator fun plus(other: Duration): Duration {\n    when {\n        this.isInfinite() -> {\n            if
(other.isFinite() || (this.rawValue xor other.rawValue >= 0))\n                return this\n            else\n
throw IllegalArgumentException("Summing infinite durations of different signs yields an undefined result.")\n
        }\n    }\n}

```

```

        other.isInfinite() -> return other\n    }\n\n    return when {\n        this.unitDiscriminator ==
other.unitDiscriminator -> {\n        val result = this.value + other.value // never overflows long, but can
overflow long63\n        when {\n            isInNanos() ->\n
durationOfNanosNormalized(result)\n            else ->\n                durationOfMillisNormalized(result)\n
        }\n    }\n    this.isInMillis() ->\n        addValuesMixedRanges(this.value, other.value)\n
else ->\n        addValuesMixedRanges(other.value, this.value)\n    }\n }\n\n private fun
addValuesMixedRanges(thisMillis: Long, otherNanos: Long): Duration {\n    val otherMillis =
nanosToMillis(otherNanos)\n    val resultMillis = thisMillis + otherMillis\n    return if (resultMillis in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n        val otherNanoRemainder
= otherNanos - millisToNanos(otherMillis)\n        durationOfNanos(millisToNanos(resultMillis) +
otherNanoRemainder)\n    } else {\n        durationOfMillis(resultMillis.coerceIn(-MAX_MILLIS,
MAX_MILLIS))\n    }\n }\n\n /**\n  * Returns a duration whose value is the difference between this and
[other] duration values.\n  * @throws IllegalArgumentException if the operation results in an undefined
value for the given arguments,\n  * e.g. when subtracting infinite durations of the same sign.\n  */\n public
operator fun minus(other: Duration): Duration = this + (-other)\n\n /**\n  * Returns a duration whose value is
this duration value multiplied by the given [scale] number.\n  * @throws IllegalArgumentException if the
operation results in an undefined value for the given arguments,\n  * e.g. when multiplying an infinite duration by
zero.\n  */\n public operator fun times(scale: Int): Duration {\n    if (isInfinite())
{\n        return when {\n            scale == 0 -> throw IllegalArgumentException("Multiplying infinite duration
by zero yields an undefined result.")\n            scale > 0 -> this\n            else -> -this\n        }\n    }\n    if
(scale == 0) return ZERO\n    val value = value\n    val result = value * scale\n    return if (isInNanos()) {\n
        if (value in (MAX_NANOS / Int.MIN_VALUE)..(-MAX_NANOS / Int.MIN_VALUE)) {\n            // can't
overflow nanos range for any scale\n            durationOfNanos(result)\n        } else {\n            if (result / scale
== value) {\n                durationOfNanosNormalized(result)\n            } else {\n                val millis =
nanosToMillis(value)\n                val remNanos = value - millisToNanos(millis)\n                val resultMillis =
millis * scale\n                val totalMillis = resultMillis + nanosToMillis(remNanos * scale)\n
                if (resultMillis / scale == millis && totalMillis xor resultMillis >= 0) {\n
                    durationOfMillis(totalMillis.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n                } else {\n                    if
(value.sign * scale.sign > 0) INFINITE else NEG_INFINITE\n                }\n            }\n        }\n    } else
{\n        if (result / scale == value) {\n            durationOfMillis(result.coerceIn(-
MAX_MILLIS..MAX_MILLIS))\n        } else {\n            if (value.sign * scale.sign > 0) INFINITE else
NEG_INFINITE\n        }\n    }\n }\n\n /**\n  * Returns a duration whose value is this duration value
multiplied by the given [scale] number.\n  * The operation may involve rounding when the result cannot be
represented exactly with a [Double] number.\n  * @throws IllegalArgumentException if the operation
results in an undefined value for the given arguments,\n  * e.g. when
multiplying an infinite duration by zero.\n  */\n public operator fun times(scale: Double): Duration {\n    val
intScale = scale.roundToInt()\n    if (intScale.toDouble() == scale) {\n        return times(intScale)\n    }\n\n    val unit = storageUnit\n    val result = toDouble(unit) * scale\n    return result.toDuration(unit)\n }\n\n /**\n  * Returns a duration whose value is this duration value divided by the given [scale] number.\n  * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n  * e.g. when dividing zero duration by zero.\n  */\n public operator fun div(scale: Int): Duration {\n    if (scale ==
0) {\n        return when {\n            isPositive() -> INFINITE\n            isNegative() -> NEG_INFINITE\n
            else -> throw IllegalArgumentException("Dividing zero duration by zero yields an undefined result.")\n
        }\n    }\n\n    if (isInNanos()) {\n        return durationOfNanos(value / scale)\n    } else {\n        if (isInfinite())\n        return this * scale.sign\n        val result = value / scale\n        if (result in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n            val rem = millisToNanos(value - (result *
scale)) / scale\n            return durationOfNanos(millisToNanos(result) + rem)\n        }\n        return

```

```

durationOfMillis(result)\n    }\n    }\n\n    /**\n     * Returns a duration whose value is this duration value divided
by the given [scale] number.\n     *\n     * @throws IllegalArgumentException if the operation results in an
undefined value for the given arguments,\n     * e.g. when dividing an infinite duration by infinity or zero duration
by zero.\n     */\n    public operator fun div(scale: Double): Duration {\n        val intScale = scale.roundToInt()\n        if (intScale.toDouble() == scale && intScale != 0) {\n            return div(intScale)\n        }\n        val unit = storageUnit\n        val result = toDouble(unit) / scale\n        return result.toDuration(unit)\n    }\n\n    /** Returns a number that is the ratio of this and [other] duration values.
*/\n    public operator fun div(other: Duration): Double {\n        val coarserUnit = maxOf(this.storageUnit,
other.storageUnit)\n        return this.toDouble(coarserUnit) / other.toDouble(coarserUnit)\n    }\n\n    /** Returns
true, if the duration value is less than zero. */\n    public fun isNegative(): Boolean = rawValue < 0\n\n    /** Returns
true, if the duration value is greater than zero. */\n    public fun isPositive(): Boolean = rawValue > 0\n\n    /**
Returns true, if the duration value is infinite. */\n    public fun isInfinite(): Boolean = rawValue ==
INFINITE.rawValue || rawValue == NEG_INFINITE.rawValue\n\n    /** Returns true, if the duration value is finite.
*/\n    public fun isFinite(): Boolean = !isInfinite()\n\n    Returns the absolute value of this value. The returned value is always non-negative. */\n    public val absoluteValue:
Duration get() = if (isNegative()) -this else this\n\n    override fun compareTo(other: Duration): Int {\n        val
compareBits = this.rawValue xor other.rawValue\n        if (compareBits < 0 || compareBits.toInt() and 1 == 0) //
different signs or same sign/same range\n            return this.rawValue.compareTo(other.rawValue) // same
sign/different ranges\n        val r = this.unitDiscriminator - other.unitDiscriminator // compare ranges\n        return if
(isNegative()) -r else r\n    }\n\n    /**\n     * Splits this duration into days, hours,
minutes, seconds, and nanoseconds and executes the given [action] with these components.\n     * The result of
[action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole number of
nanoseconds in this duration, and its absolute value
is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration, and its
absolute value is less than 60;\n     * - `minutes` represents the whole number of minutes in this duration, and its
absolute value is less than 60;\n     * - `hours` represents the whole number of hours in this duration, and its absolute
value is less than 24;\n     * - `days` represents the whole number of days in this duration.\n     *\n     * Infinite
durations are represented as either [Long.MAX_VALUE] days, or [Long.MIN_VALUE] days (depending on the
sign of infinity),\n     * and zeroes in the lower components.\n     */\n    public inline fun <T> toComponents(action:
(days: Long, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action,
InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeDays, hoursComponent, minutesComponent,
secondsComponent, nanosecondsComponent)\n    }\n\n    /**\n     * Splits this
duration into hours, minutes, seconds, and nanoseconds and executes the given [action] with these components.\n     *
The result of [action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole
number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n     * - `seconds`
represents the whole number of seconds in this duration, and its absolute value is less than 60;\n     * - `minutes`
represents the whole number of minutes in this duration, and its absolute value is less than 60;\n     * - `hours`
represents the whole number of hours in this duration.\n     *\n     * Infinite durations are represented as either
[Long.MAX_VALUE] hours, or [Long.MIN_VALUE] hours (depending on the sign of infinity),\n     * and zeroes
in the lower components.\n     */\n    public inline fun <T> toComponents(action: (hours: Long, minutes: Int,
seconds: Int, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action,
InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeHours, minutesComponent, secondsComponent,
nanosecondsComponent)\n    }\n\n    /**\n     * Splits this duration into minutes, seconds, and nanoseconds and
executes the given [action] with these components.\n     * The result of [action] is returned as the result of this
function.\n     *\n     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute
value is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration, and its
absolute value is less than 60;\n     * - `minutes` represents the whole number of minutes in this duration.\n     *\n     *
Infinite durations are represented as either [Long.MAX_VALUE] minutes, or [Long.MIN_VALUE] minutes

```

```

(depending on the sign of infinity),\n * and zeroes in the lower components.\n */\n public inline fun <T>
toComponents(action: (minutes: Long, seconds: Int, nanoseconds: Int) -> T): T {\n
    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n    return action(inWholeMinutes,
secondsComponent, nanosecondsComponent)\n }\n\n /**\n * Splits this duration into seconds, and
nanoseconds and executes the given [action] with these components.\n * The result of [action] is returned as the
result of this function.\n *\n * - `nanoseconds` represents the whole number of nanoseconds in this duration,
and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the whole number of seconds in this
duration.\n *\n * Infinite durations are represented as either [Long.MAX_VALUE] seconds, or
[Long.MIN_VALUE] seconds (depending on the sign of infinity),\n * and zero nanoseconds.\n */\n public
inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T {\n    contract {
callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n    return action(inWholeSeconds,
nanosecondsComponent)\n
}\n\n @PublishedApi\n internal val hoursComponent: Int\n    get() = if (isInfinite()) 0 else (inWholeHours
% 24).toInt()\n\n @PublishedApi\n internal val minutesComponent: Int\n    get() = if (isInfinite()) 0 else
(inWholeMinutes % 60).toInt()\n\n @PublishedApi\n internal val secondsComponent: Int\n    get() = if
(isInfinite()) 0 else (inWholeSeconds % 60).toInt()\n\n @PublishedApi\n internal val nanosecondsComponent:
Int\n    get() = when {\n        isInfinite() -> 0\n        isInMillis() -> millisToNanos(value % 1_000).toInt()\n
        else -> (value % 1_000_000_000).toInt()\n    }\n\n // conversion to units\n\n /**\n * Returns the value
of this duration expressed as a [Double] number of the specified [unit].\n *\n * The operation may involve
rounding when the result cannot be represented exactly with a [Double] number.\n *\n * An infinite duration
value is converted either to [Double.POSITIVE_INFINITY]
or [Double.NEGATIVE_INFINITY] depending on its sign.\n */\n public fun toDouble(unit: DurationUnit):
Double {\n    return when (rawValue) {\n        INFINITE.rawValue -> Double.POSITIVE_INFINITY\n
NEG_INFIMATE.rawValue -> Double.NEGATIVE_INFINITY\n        else -> {\n            // TODO: whether it's
ok to convert to Double before scaling\n            convertDurationUnit(value.toDouble(), storageUnit, unit)\n
        }\n    }\n\n /**\n * Returns the value of this duration expressed as a [Long] number of the specified
[unit].\n *\n * If the result doesn't fit in the range of [Long] type, it is coerced into that range:\n * -
[Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if
it's greater than `Long.MAX_VALUE`.\n *\n * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public fun
toLong(unit: DurationUnit): Long {\n    return when (rawValue) {\n        INFINITE.rawValue ->
Long.MAX_VALUE\n        NEG_INFIMATE.rawValue -> Long.MIN_VALUE\n        else ->
convertDurationUnit(value, storageUnit, unit)\n    }\n\n /**\n * Returns the value of this duration
expressed as an [Int] number of the specified [unit].\n *\n * If the result doesn't fit in the range of [Int] type, it
is coerced into that range:\n * - [Int.MIN_VALUE] is returned if it's less than `Int.MIN_VALUE`,\n * -
[Int.MAX_VALUE] is returned if it's greater than `Int.MAX_VALUE`.\n *\n * An infinite duration value is
converted either to [Int.MAX_VALUE] or [Int.MIN_VALUE] depending on its sign.\n */\n public fun
toInt(unit: DurationUnit): Int =\n    toLong(unit).coerceIn(Int.MIN_VALUE.toInt(),
Int.MAX_VALUE.toInt()).toInt()\n\n /** The value of this duration expressed as a [Double] number of days.
*/\n @ExperimentalTime\n @Deprecated("Use
inWholeDays property instead or convert toDouble(DAYS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.DAYS)"))\n public val inDays: Double\n    get() =
toDouble(DurationUnit.DAYS)\n\n /** The value of this duration expressed as a [Double] number of hours. */\n
@ExperimentalTime\n @Deprecated("Use inWholeHours property instead or convert toDouble(HOURS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.HOURS)"))\n public val inHours: Double\n
    get() = toDouble(DurationUnit.HOURS)\n\n /** The value of this duration expressed as a [Double] number of
minutes. */\n @ExperimentalTime\n @Deprecated("Use inWholeMinutes property instead or convert
toDouble(MINUTES) if a double value is required.", ReplaceWith("toDouble(DurationUnit.MINUTES)"))\n

```

```

public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)\n\n /** The value of this duration
expressed as a [Double] number of seconds. */\n @ExperimentalTime\n
@Deprecated("Use inWholeSeconds property instead or convert toDouble(SECONDS) if a double value is
required.", ReplaceWith("toDouble(DurationUnit.SECONDS)"))\n public val inSeconds: Double get() =
toDouble(DurationUnit.SECONDS)\n\n /** The value of this duration expressed as a [Double] number of
milliseconds. */\n @ExperimentalTime\n @Deprecated("Use inWholeMilliseconds property instead or convert
toDouble(MILLISECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MILLISECONDS)"))\n public val inMilliseconds: Double get() =
toDouble(DurationUnit.MILLISECONDS)\n\n /** The value of this duration expressed as a [Double] number of
microseconds. */\n @ExperimentalTime\n @Deprecated("Use inWholeMicroseconds property instead or
convert toDouble(MICROSECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MICROSECONDS)"))\n public val inMicroseconds: Double get() =
toDouble(DurationUnit.MICROSECONDS)\n\n
/** The value of this duration expressed as a [Double] number of nanoseconds. */\n @ExperimentalTime\n
@Deprecated("Use inWholeNanoseconds property instead or convert toDouble(NANOSECONDS) if a double
value is required.", ReplaceWith("toDouble(DurationUnit.NANOSECONDS)"))\n public val inNanoseconds:
Double get() = toDouble(DurationUnit.NANOSECONDS)\n\n\n /**\n * The value of this duration expressed as
a [Long] number of days.\n * \n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n * \n public val inWholeDays: Long\n get() =
toLong(DurationUnit.DAYS)\n\n\n /**\n * The value of this duration expressed as a [Long] number of hours.\n
*\n * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending
on its sign.\n * \n public val inWholeHours: Long\n get() = toLong(DurationUnit.HOURS)\n\n\n /**\n *
The value of this
duration expressed as a [Long] number of minutes.\n * \n * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n * \n public val inWholeMinutes:
Long\n get() = toLong(DurationUnit.MINUTES)\n\n\n /**\n * The value of this duration expressed as a
[Long] number of seconds.\n * \n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n * \n public val inWholeSeconds: Long\n get() =
toLong(DurationUnit.SECONDS)\n\n\n /**\n * The value of this duration expressed as a [Long] number of
milliseconds.\n * \n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n * \n public val inWholeMilliseconds: Long\n get() {\n
return if (isInMillis() && isFinite()) value else toLong(DurationUnit.MILLISECONDS)\n } \n\n\n /**\n *
The value of this
duration expressed as a [Long] number of microseconds.\n * \n * If the result doesn't fit in the range of [Long]
type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE` ,\n
* - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE` .\n * \n * An infinite duration
value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n * \n public
val inWholeMicroseconds: Long\n get() = toLong(DurationUnit.MICROSECONDS)\n\n\n /**\n * The value
of this duration expressed as a [Long] number of nanoseconds.\n * \n * If the result doesn't fit in the range of
[Long] type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE` ,\n * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE` .\n * \n
* An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending
on its sign.\n * \n public val inWholeNanoseconds: Long\n get() {\n val value = value\n
return when {\n isInNanos() -> value\n value > Long.MAX_VALUE / NANOS_IN_MILLIS ->
Long.MAX_VALUE\n value < Long.MIN_VALUE / NANOS_IN_MILLIS -> Long.MIN_VALUE\n
else -> millisToNanos(value)\n } \n } \n\n // shortcuts\n\n\n /**\n * Returns the value of this
duration expressed as a [Long] number of nanoseconds.\n * \n * If the value doesn't fit in the range of [Long]
type, it is coerced into that range, see the conversion [Double.toLong] for details.\n * \n * The range of

```

```

durations that can be expressed as a `Long` number of nanoseconds is approximately 1292 years.
    @ExperimentalTime
    @Deprecated("Use inWholeNanoseconds property instead.")
    ReplaceWith("this.inWholeNanoseconds")
    public fun toLongNanoseconds(): Long =
        inWholeNanoseconds

    /**
     * Returns the value of this duration expressed as a [Long] number of milliseconds.
     * The value is coerced to the range of [Long] type, if it doesn't fit in that range, see the conversion [Double.toLong] for details.
     * The range of durations that can be expressed as a `Long` number of milliseconds is approximately 1292 million years.
     * @ExperimentalTime
     * @Deprecated("Use inWholeMilliseconds property instead.", ReplaceWith("this.inWholeMilliseconds"))
     * public fun toLongMilliseconds(): Long =
     * inWholeMilliseconds

    /**
     * Returns a string representation of this duration value expressed as a combination of numeric components, each in its own unit.
     * Each component is a number followed by the unit abbreviated name: `d`, `h`, `m`, `s`, `5h`, `1d 12h`, `1h 0m 30.340s`.
     * The last component, usually seconds, can be a number with a fractional part.
     * If the duration is less than a second, it is represented as a single number with one of sub-second units: `ms` (milliseconds), `us` (microseconds), or `ns` (nanoseconds): `140.884ms`, `500us`, `24ns`.
     * A negative duration is prefixed with `-` sign and, if it consists of multiple components, surrounded with parentheses: `-12m` and `-(1h 30m)`.
     * Special cases:
     * - an infinite duration is formatted as `"Infinity"` or `"-Infinity"` without a unit.
     * It's recommended to use [toIsoString] that uses more strict ISO-8601 format instead of this `toString` when you want to convert a duration to a string in cases of serialization, interchange, etc.
     * @sample samples.time.Durations.toStringDefault
     * override fun toString(): String = when (rawValue) {
     *     0L -> "0s"
     *     INFINITE.rawValue -> "Infinity"
     *     NEG_INFINITE.rawValue -> "-Infinity"
     *     else -> {
     *         val isNegative = isNegative()
     *         buildString {
     *             if (isNegative) append('-')
     *             absoluteValue.toComponents { days, hours, minutes, seconds, nanoseconds ->
     *                 val hasDays = days != 0L
     *                 val hasHours = hours != 0L
     *                 val hasMinutes = minutes != 0L
     *                 val hasSeconds = seconds != 0 || nanoseconds != 0L
     *                 var components = 0L
     *                 if (hasDays) {
     *                     append(days).append('d')
     *                     components++
     *                 }
     *                 if (hasHours || (hasDays && (hasMinutes || hasSeconds))) {
     *                     if (components++ > 0) append(' ')
     *                     append(hours).append('h')
     *                 }
     *                 if (hasMinutes || (hasSeconds && (hasHours || hasDays))) {
     *                     if (components++ > 0) append(' ')
     *                     append(minutes).append('m')
     *                 }
     *                 if (hasSeconds) {
     *                     if (components++ > 0) append(' ')
     *                     when (seconds != 0 || hasDays || hasHours || hasMinutes ->
     *                         appendFractional(seconds, nanoseconds, 9, "s", isoZeroes = false)
     *                         nanoseconds >= 1_000_000 ->
     *                             appendFractional(nanoseconds / 1_000_000, nanoseconds % 1_000_000, 6, "ms", isoZeroes = false)
     *                         nanoseconds >= 1_000 ->
     *                             appendFractional(nanoseconds / 1_000, nanoseconds % 1_000, 3, "us", isoZeroes = false)
     *                         else ->
     *                             append(nanoseconds).append("ns")
     *                     }
     *                 }
     *                 if (isNegative && components > 1) insert(1, '(').append(')')
     *             }
     *         }
     *     }
     * }
     * private fun
     * StringBuilder.appendFractional(whole: Int, fractional: Int, fractionalSize: Int, unit: String, isoZeroes: Boolean) {
     *     append(whole)
     *     if (fractional != 0) {
     *         append('.')
     *         val fracString = fractional.toString().padStart(fractionalSize, '0')
     *         val nonZeroDigits = fracString.indexOfLast { it != '0' } + 1
     *         when {
     *             !isoZeroes && nonZeroDigits < 3 -> appendRange(fracString, 0, nonZeroDigits)
     *             else -> appendRange(fracString, 0, ((nonZeroDigits + 2) / 3) * 3)
     *         }
     *         append(unit)
     *     }
     * }
     * /**
     * Returns a string representation of this duration value expressed in the given [unit] and formatted with the specified [decimals] number of digits after decimal point.
     * Special cases:
     * - an infinite duration is formatted as `"Infinity"` or `"-Infinity"` without a unit.
     * @param decimals the number of digits after decimal point

```


to show. The value must be non-negative.

- * No more than 12 decimals will be shown, even if a larger number is requested.
- * @return the value of duration in the specified [unit] followed by that unit abbreviated name: `d`, `h`, `m`, `s`, `ms`, `us`, or `ns`.
- * @throws IllegalArgumentException if [decimals] is less than zero.
- * @sample samples.time.Durations.toStringDecimals

```

public fun toString(unit: DurationUnit, decimals: Int = 0): String {
    require(decimals >= 0) { "decimals must be not negative, but was $decimals" }
    val number = toDouble(unit)
    if (number.isInfinite()) return number.toString()
    return formatToExactDecimals(number, decimals.coerceAtMost(12)) + unit.shortName()
}

```

/* Returns an ISO-8601 based string representation of this duration. The returned value is presented in the format `PTmMs.fS`, where `h`, `m`, `s` are the integer components of this duration (see [toComponents]) and `f` is a fractional part of second. Depending on the roundness of the value the fractional part can be formatted with either 0, 3, 6, or 9 decimal digits. The infinite duration is represented as `PT999999999999999H` which is larger than any possible finite duration in Kotlin. Negative durations are indicated with the sign `-` in the beginning of the returned string, for example, `-PT5M30S`.

- * @sample samples.time.Durations.toIsoString

```

public fun toIsoString(): String = buildString {
    if (isNegative()) append('-')
    append("PT")
    this@Duration.absoluteValue.toComponents { hours, minutes, seconds, nanoseconds ->
        @Suppress("NAME_SHADOWING")
        var hours = hours
        if (isInfinite()) { // use large enough value instead of Long.MAX_VALUE
            hours = 9_999_999_999_999
        }
        val hasHours = hours != 0L
        val hasSeconds = seconds != 0 || nanoseconds != 0
        val hasMinutes = minutes != 0 || (hasSeconds && hasHours)
        if (hasHours) {
            append(hours).append('H')
        }
        if (hasMinutes) {
            append(minutes).append('M')
        }
        if (hasSeconds || (!hasHours && !hasMinutes)) {
            appendFractional(seconds, nanoseconds, 9, "S", isoZeroes = true)
        }
    }
}

```

/* Returns a [Duration] equal to this [Int] number of the specified [unit].

```

@SinceKotlin("1.6")@WasExperimental(ExperimentalTime::class)
public fun Int.toDuration(unit: DurationUnit): Duration {
    return if (unit <= DurationUnit.SECONDS) {
        durationOfNanos(convertDurationUnitOverflow(this.toLong(), unit, DurationUnit.NANOSECONDS))
    } else {
        toLong().toDuration(unit)
    }
}

```

/* Returns a [Duration] equal to this [Long] number of the specified [unit].

```

@SinceKotlin("1.6")@WasExperimental(ExperimentalTime::class)
public fun Long.toDuration(unit: DurationUnit): Duration {
    val maxNsInUnit = convertDurationUnitOverflow(MAX_NANOS, DurationUnit.NANOSECONDS, unit)
    if (this in -maxNsInUnit..maxNsInUnit) {
        return durationOfNanos(convertDurationUnitOverflow(this, unit, DurationUnit.NANOSECONDS))
    } else {
        val millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS)
        return durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))
    }
}

```

/* Returns a [Duration] equal to this [Double] number of the specified [unit]. Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.

- * @throws IllegalArgumentException if this `Double` value is `NaN`.

```

@SinceKotlin("1.6")@WasExperimental(ExperimentalTime::class)
public fun Double.toDuration(unit: DurationUnit): Duration {
    val valueInNs = convertDurationUnit(this, unit, DurationUnit.NANOSECONDS)
    require(!valueInNs.isNaN()) { "Duration value cannot be NaN." }
    val nanos = valueInNs.roundToLong()
    return if (nanos in -MAX_NANOS..MAX_NANOS) {
        durationOfNanos(nanos)
    } else {
        val millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS).roundToLong()
        durationOfMillisNormalized(millis)
    }
}

```

/* Returns a [Duration] equal to this [Int] number of nanoseconds.

```

@SinceKotlin("1.3")@ExperimentalTime@Deprecated("Use 'Int.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))
@DeprecatedSinceKotlin(warningSince = "1.5")
public val

```

```

Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/** Returns a [Duration] equal to
this [Long] number of nanoseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"kotlin.time.Duration.Companion.nanoseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/**\n * Returns a [Duration] equal to this
[Double] number of nanoseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.nanoseconds' extension property
from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"kotlin.time.Duration.Companion.nanoseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/** Returns a [Duration] equal to
this [Int] number of microseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Int.microseconds' extension property from Duration.Companion instead.", ReplaceWith("this.microseconds",
"kotlin.time.Duration.Companion.microseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a [Duration] equal to this
[Long] number of microseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.microseconds' extension property from Duration.Companion instead.", ReplaceWith("this.microseconds",
"kotlin.time.Duration.Companion.microseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/**\n * Returns a [Duration] equal to
this [Double] number of microseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.microseconds' extension
property from Duration.Companion
instead.", ReplaceWith("this.microseconds",
"kotlin.time.Duration.Companion.microseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a [Duration] equal to
this [Int] number of milliseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Int.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Long] number of milliseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/**\n * Returns a [Duration] equal to
this [Double] number of milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.milliseconds' extension property
from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Int] number of seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.seconds'
extension property from Duration.Companion instead.", ReplaceWith("this.seconds",
"kotlin.time.Duration.Companion.seconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal
to this [Long] number of seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.seconds' extension property from Duration.Companion instead.", ReplaceWith("this.seconds",
"kotlin.time.Duration.Companion.seconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of seconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n *\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.seconds' extension property from
Duration.Companion instead.", ReplaceWith("this.seconds",

```

```

\kotlin.time.Duration.Companion.seconds\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Int]
number of minutes. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.minutes' extension
property
from Duration.Companion instead.\", ReplaceWith(\"this.minutes\"),
\kotlin.time.Duration.Companion.minutes\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Long] number of
minutes. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.minutes' extension property
from Duration.Companion instead.\", ReplaceWith(\"this.minutes\"),
\kotlin.time.Duration.Companion.minutes\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of minutes.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.minutes' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.minutes\"),
\kotlin.time.Duration.Companion.minutes\)\n@DeprecatedSinceKotlin(warningSince
= \"1.5\")\npublic val Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration]
equal to this [Int] number of hours. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Int.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\"),
\kotlin.time.Duration.Companion.hours\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Int.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Long] number of
hours. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.hours' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.hours\"),
\kotlin.time.Duration.Companion.hours\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Long.hours get() = toDuration(DurationUnit.HOURS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of hours.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Double.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\"),
\kotlin.time.Duration.Companion.hours\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Double.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Int] number of
days. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.days' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.days\"),
\kotlin.time.Duration.Companion.days\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val Int.days
get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Long] number of days.
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.days' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.days\"),
\kotlin.time.Duration.Companion.days\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic
val Long.days get() = toDuration(DurationUnit.DAYS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of days.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.days' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.days\"),
\kotlin.time.Duration.Companion.days\)\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic val
Double.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value is the specified
[duration] value multiplied by this number.
*\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Int.times(duration: Duration): Duration = duration * this\n\n/**\n * Returns a duration whose
value is the specified [duration] value multiplied by this number.\n *\n * The operation may involve rounding when
the result cannot be represented exactly with

```

```

a [Double] number.\n *\n * @throws IllegalArgumentException if the operation results in a `NaN` value.\n
*\n @SinceKotlin("1.6")\n @WasExperimental(ExperimentalTime::class)\n @kotlin.internal.InlineOnly\n public
inline operator fun Double.times(duration: Duration): Duration = duration * this\n\n\n private fun
parseDuration(value: String, strictIso: Boolean): Duration {\n    var length = value.length\n    if (length == 0) throw
IllegalArgumentException("The string is empty")\n    var index = 0\n    var result = Duration.ZERO\n    val
infinityString = "Infinity"\n    when (value[index]) {\n        '+', '-' -> index++\n    }\n    val hasSign = index > 0\n    val
isNegative = hasSign && value.startsWith('-')\n    when {\n        length <= index ->\n            throw
IllegalArgumentException("No components")\n        value[index] == 'P' -> {\n            if (++index == length) throw
IllegalArgumentException()\n            val nonDigitSymbols = "+-." \n            var isTimeComponent
= false\n            var prevUnit: DurationUnit? = null\n            while (index < length) {\n                if (value[index] ==
'T') {\n                    if (isTimeComponent || ++index == length) throw IllegalArgumentException()\n                    isTimeComponent = true\n                    continue\n                }\n                val component =
value.substringWhile(index) { it in '0'..'9' || it in nonDigitSymbols }\n                if (component.isEmpty()) throw
IllegalArgumentException()\n                index += component.length\n                val unitChar =
value.getOrElse(index) { throw IllegalArgumentException("Missing unit for value $component") }\n                index++\n                val unit = durationUnitByIsoChar(unitChar, isTimeComponent)\n                if (prevUnit != null
&& prevUnit <= unit) throw IllegalArgumentException("Unexpected order of duration components")\n                prevUnit = unit\n                val dotIndex = component.indexOf('.')\n                if (unit == DurationUnit.SECONDS && dotIndex > 0) {\n                    val whole =
component.substring(0, dotIndex)\n                    result += parseOverLongIsoComponent(whole).toDuration(unit)\n                    result += component.substring(dotIndex).toDouble().toDuration(unit)\n                } else {\n                    result += parseOverLongIsoComponent(component).toDuration(unit)\n                }\n            }\n            strictIso
->\n                throw IllegalArgumentException()\n            value.regionMatches(index, infinityString, 0, length =
maxOf(length - index, infinityString.length), ignoreCase = true) -> {\n                result = Duration.INFINITE\n            }\n            else -> {\n                // parse default string format\n                var prevUnit: DurationUnit? = null\n                var
afterFirst = false\n                var allowSpaces = !hasSign\n                if (hasSign && value[index] == '(' && value.last()
== ')') {\n                    allowSpaces
= true\n                    if (++index == --length) throw IllegalArgumentException("No components")\n                }\n                while (index < length) {\n                    if (afterFirst && allowSpaces) {\n                        index = value.skipWhile(index) {
it == ' ' }\n                    }\n                    afterFirst = true\n                    val component = value.substringWhile(index) { it in
'0'..'9' || it == '.' }\n                    if (component.isEmpty()) throw IllegalArgumentException()\n                    index +=
component.length\n                    val unitName = value.substringWhile(index) { it in 'a'..'z' }\n                    index +=
unitName.length\n                    val unit = durationUnitByShortName(unitName)\n                    if (prevUnit != null &&
prevUnit <= unit) throw IllegalArgumentException("Unexpected order of duration components")\n                    prevUnit = unit\n                    val dotIndex = component.indexOf('.')\n                    if (dotIndex > 0) {\n                        val whole = component.substring(0, dotIndex)\n                        result += whole.toLong().toDuration(unit)\n                        result += component.substring(dotIndex).toDouble().toDuration(unit)\n                    }\n                    if (index < length) throw
IllegalArgumentException("Fractional component must be last")\n                } else {\n                    result +=
component.toLong().toDuration(unit)\n                }\n            }\n            }\n            }\n            }\n            return if (isNegative) -result else
result\n\n\n private fun parseOverLongIsoComponent(value: String): Long {\n    val length = value.length\n    var
startIndex = 0\n    if (length > 0 && value[0] in "+-") startIndex++\n    if ((length - startIndex) > 16 &&
(startIndex..value.lastIndex).all { value[it] in '0'..'9' }) {\n        // all chars are digits, but more than
ceiling(log10(MAX_MILLIS / 1000)) of them\n        return if (value[0] == '-') Long.MIN_VALUE else
Long.MAX_VALUE\n    }\n    // TODO: replace with just toLong after
min JDK becomes 8\n    return if (value.startsWith("+")) value.drop(1).toLong() else
value.toLong()\n\n\n private inline fun String.substringWhile(startIndex: Int, predicate: (Char) -> Boolean):
String =\n    substring(startIndex, skipWhile(startIndex, predicate))\n\n private inline fun
String.skipWhile(startIndex: Int, predicate: (Char) -> Boolean): Int {\n    var i = startIndex\n    while (i < length &&

```

```

predicate(this[i])) i++\n    return i}\n\n\n\n\n// The ranges are chosen so that they are:\n// - symmetric relative to
zero: this greatly simplifies operations with sign, e.g. unaryMinus and minus.\n// - non-overlapping, but adjacent:
the first value that doesn't fit in nanos range, can be exactly represented in millis.\n\n\ninternal const val
NANOS_IN_MILLIS = 1_000_000\n// maximum number duration can store in nanosecond range\n\ninternal const
val MAX_NANOS = Long.MAX_VALUE / 2 / NANOS_IN_MILLIS * NANOS_IN_MILLIS - 1 // ends in
..._999_999\n// maximum number duration can store
in millisecond range, also encodes an infinite value\n\ninternal const val MAX_MILLIS = Long.MAX_VALUE /
2\n// MAX_NANOS expressed in milliseconds\n\nprivate const val MAX_NANOS_IN_MILLIS = MAX_NANOS /
NANOS_IN_MILLIS\n\nprivate fun nanosToMillis(nanos: Long): Long = nanos / NANOS_IN_MILLIS\n\nprivate
fun millisToNanos(millis: Long): Long = millis * NANOS_IN_MILLIS\n\nprivate fun
durationOfNanos(normalNanos: Long) = Duration(normalNanos shl 1)\n\nprivate fun durationOfMillis(normalMillis:
Long) = Duration((normalMillis shl 1) + 1)\n\nprivate fun durationOf(normalValue: Long, unitDiscriminator: Int) =
Duration((normalValue shl 1) + unitDiscriminator)\n\nprivate fun durationOfNanosNormalized(nanos: Long) =\n    if
(nanos in -MAX_NANOS..MAX_NANOS) {\n        durationOfNanos(nanos)\n    } else {\n
durationOfMillis(nanosToMillis(nanos))\n    }\n\nprivate fun durationOfMillisNormalized(millis: Long) =\n    if
(millis in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n
durationOfNanos(millisToNanos(millis))\n    } else {\n        durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n    }\n\n\ninternal expect val
durationAssertionsEnabled: Boolean\n\n\ninternal expect fun formatToExactDecimals(value: Double, decimals: Int):
String\n\n\ninternal expect fun formatUpToDecimals(value: Double, decimals: Int): String", "/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmName("UnsignedKt")\n\npackage kotlin\n\n\n@PublishedApi\n\n\ninternal fun uintCompare(v1:
Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)\n\n\n@PublishedApi\n\n\ninternal fun
ulongCompare(v1: Long, v2: Long): Int = (v1 xor Long.MIN_VALUE).compareTo(v2 xor
Long.MIN_VALUE)\n\n\n@PublishedApi\n\n\ninternal fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() /
v2.toLong()).toUInt()\n\n\n@PublishedApi\n\n\ninternal fun uintRemainder(v1:
UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()\n\n\n// Division and remainder are based on Guava's
UnsignedLongs implementation\n\n// Copyright 2011 The Guava Authors\n\n\n@PublishedApi\n\n\ninternal fun
ulongDivide(v1: ULong, v2: ULong): ULong {\n    val dividend = v1.toLong()\n    val divisor = v2.toLong()\n    if
(divisor < 0) { // i.e., divisor >= 2^63:\n        return if (v1 < v2) ULong(0) else ULong(1)\n    }\n\n    // Optimization
- use signed division if both dividend and divisor < 2^63\n    if (dividend >= 0) {\n        return ULong(dividend /
divisor)\n    }\n\n    // Otherwise, approximate the quotient, check, and correct if necessary.\n    val quotient =
((dividend ushr 1) / divisor) shl 1\n    val rem = dividend - quotient * divisor\n    return ULong(quotient + if
(ULong(rem) >= ULong(divisor)) 1 else 0)\n}\n\n\n@PublishedApi\n\n\ninternal fun ulongRemainder(v1: ULong, v2:
ULong): ULong {\n    val dividend = v1.toLong()\n    val divisor = v2.toLong()\n    if (divisor < 0)
{ // i.e., divisor >= 2^63:\n        return if (v1 < v2) {\n            v1 // dividend < divisor\n        } else {\n
v1 - v2 // dividend >= divisor\n        }\n    }\n\n    // Optimization - use signed modulus if both dividend and divisor < 2^63\n
if (dividend >= 0) {\n        return ULong(dividend % divisor)\n    }\n\n    // Otherwise, approximate the quotient,
check, and correct if necessary.\n    val quotient = ((dividend ushr 1) / divisor) shl 1\n    val rem = dividend -
quotient * divisor\n    return ULong(rem - if (ULong(rem) >= ULong(divisor)) divisor else
0)\n}\n\n\n@PublishedApi\n\n\ninternal fun doubleToUInt(v: Double): UInt = when {\n    v.isNaN() -> 0u\n    v <=
UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE\n    v >= UInt.MAX_VALUE.toDouble() ->
UInt.MAX_VALUE\n    v <= Int.MAX_VALUE -> v.toInt().toUInt()\n    else -> (v -
Int.MAX_VALUE).toInt().toUInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v <
UInt.MAX_VALUE\n}\n\n\n@PublishedApi\n\n\ninternal fun doubleToULong(v:
Double): ULong = when {\n    v.isNaN() -> 0u\n    v <= ULong.MIN_VALUE.toDouble() ->
ULong.MIN_VALUE\n    v >= ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE\n    v <

```

```

Long.MAX_VALUE -> v.toLong().toULong()\n\n // Real values from Long.MAX_VALUE to
(Long.MAX_VALUE + 1) are not representable in Double, so don't handle them.\n else -> (v -
9223372036854775808.0).toLong().toULong() + 9223372036854775808uL // Long.MAX_VALUE + 1 < v <
ULong.MAX_VALUE\n}\n\n\n@PublishedApi\ninternal fun uintToDouble(v: Int): Double = (v and
Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n\n\n@PublishedApi\ninternal fun
ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\n\n\ninternal fun
ulongToString(v: Long): String = ulongToString(v, 10)\n\n\ninternal fun ulongToString(v: Long, base: Int): String {\n
if (v >= 0) return v.toString(base)\n\n var quotient = ((v ushr 1) / base) shl 1\n var rem = v - quotient * base\n
if (rem >= base) {\n rem
-= base\n quotient += 1\n }\n return quotient.toString(base) + rem.toString(base)\n}\n\n", "/*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\n/**\n * Given an [iterator] function constructs an [Iterable] instance that returns values through
the [Iterator]\n * provided by that function.\n * @sample samples.collections.Iterables.Building.iterable\n
*\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T>
= object : Iterable<T> {\n override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * A wrapper over another
[Iterable] (or any other object that can produce an [Iterator]) that returns\n * an indexing iterator.\n *\n\ninternal class
IndexingIterable<out
T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {\n override fun iterator():
Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())\n}\n\n\n/**\n * Returns the size of this iterable if
it is known, or `null` otherwise.\n *\n\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrNull(): Int? =
if (this is Collection<*>) this.size else null\n\n\n/**\n * Returns the size of this iterable if it is known, or the specified
[default] value otherwise.\n *\n\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrDefault(default:
Int): Int = if (this is Collection<*>) this.size else default\n\n\n/**\n * Returns a single list of all elements from all
collections in the given collection.\n * @sample samples.collections.Iterables.Operations.flattenIterable\n *\n\npublic
fun <T> Iterable<Iterable<T>>.flatten(): List<T> {\n val result = ArrayList<T>()\n for (element in this) {\n
result.addAll(element)\n }\n return
result\n}\n\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this
collection,\n * *second* list is built from the second values of each pair from this collection.\n * @sample
samples.collections.Iterables.Operations.unzipIterable\n *\n\npublic fun <T, R> Iterable<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n val expectedSize = collectionSizeOrDefault(10)\n val listT =
ArrayList<T>(expectedSize)\n val listR = ArrayList<R>(expectedSize)\n for (pair in this) {\n
listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n", "/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\npackage
kotlin.sequences\n\nimport kotlin.random.Random\n\n\n/**\n * Given an [iterator] function constructs a [Sequence] that returns values through the [Iterator]\n * provided by that
function.\n * The values are evaluated lazily, and the sequence is potentially infinite.\n *\n\n * @sample
samples.collections.Sequences.Building.sequenceFromIterator\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence(crossinline iterator: () -> Iterator<T>): Sequence<T> = object : Sequence<T> {\n override fun
iterator(): Iterator<T> = iterator()\n}\n\n\n/**\n * Creates a sequence that returns all elements from this iterator. The
sequence is constrained to be iterated only once.\n *\n\n * @sample
samples.collections.Sequences.Building.sequenceFromIterator\n *\n\npublic fun <T> Iterator<T>.asSequence():
Sequence<T> = Sequence { this }.constrainOnce()\n\n\n/**\n * Creates a sequence that returns the specified values.\n\n
*\n\n * @sample samples.collections.Sequences.Building.sequenceOfValues\n *\n\npublic fun <T> sequenceOf(vararg
elements: T): Sequence<T> = if (elements.isEmpty())

```

```

emptySequence() else elements.asSequence()\n\n/**\n * Returns an empty sequence.\n */\npublic fun <T>
emptySequence(): Sequence<T> = EmptySequence\n\nprivate object EmptySequence : Sequence<Nothing>,
DropTakeSequence<Nothing> {\n    override fun iterator(): Iterator<Nothing> = EmptyIterator\n    override fun
drop(n: Int) = EmptySequence\n    override fun take(n: Int) = EmptySequence\n}\n\n/**\n * Returns this sequence if
it's not `null` and the empty sequence otherwise.\n */\n @sample
samples.collections.Sequences.Usage.sequenceOrEmpty\n
*\n @SinceKotlin("1.3")\n @kotlin.internal.InlineOnly\n public inline fun <T> Sequence<T>?.orEmpty():
Sequence<T> = this ?: emptySequence()\n\n/**\n * Returns a sequence that iterates through the elements either of
this sequence\n * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.\n
*\n * @sample samples.collections.Sequences.Usage.sequenceIfEmpty\n */\n @SinceKotlin("1.3")\n public fun
<T>
Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {\n    val iterator =
this@ifEmpty.iterator()\n    if (iterator.hasNext()) {\n        yieldAll(iterator)\n    } else {\n
yieldAll(defaultValue())\n    }\n}\n\n/**\n * Returns a sequence of all elements from all sequences in this
sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Sequences.Transformations.flattenSequenceOfSequences\n */\n public fun <T>
Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }\n\n/**\n * Returns a sequence of all
elements from all iterables in this sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n *
@sample samples.collections.Sequences.Transformations.flattenSequenceOfLists\n
*\n @kotlin.jvm.JvmName("flattenSequenceOfIterable")\n public fun <T> Sequence<Iterable<T>>.flatten():
Sequence<T> = flatten { it.iterator() }\n\nprivate fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>):
Sequence<R> {\n    if (this is TransformingSequence<*, *>) {\n        return (this as TransformingSequence<*,
T>).flatten(iterator)\n    }\n    return FlatteningSequence(this, { it }, iterator)\n}\n\n/**\n * Returns a pair of lists,
where\n * *first* list is built from the first values of each pair from this sequence,\n * *second* list is built from the
second values of each pair from this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Sequences.Transformations.zip\n */\n public fun <T, R> Sequence<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n    val listT = ArrayList<T>()\n    val listR = ArrayList<R>()\n    for (pair in this) {\n
listT.add(pair.first)\n        listR.add(pair.second)\n    }\n    return listT to listR\n}\n\n/**\n * Returns a sequence that
yields elements of this sequence randomly shuffled.\n *\n * Note that every iteration of the sequence returns
elements in a different order.\n *\n * The operation is _intermediate_
and _stateful_.\n */\n @SinceKotlin("1.4")\n public fun <T> Sequence<T>.shuffled(): Sequence<T> =
shuffled(Random)\n\n/**\n * Returns a sequence that yields elements of this sequence randomly shuffled\n * using
the specified [random] instance as the source of randomness.\n *\n * Note that every iteration of the sequence
returns elements in a different order.\n *\n * The operation is _intermediate_ and _stateful_.\n
*\n @SinceKotlin("1.4")\n public fun <T> Sequence<T>.shuffled(random: Random): Sequence<T> =
sequence<T> {\n    val buffer = toMutableList()\n    while (buffer.isNotEmpty()) {\n        val j =
random.nextInt(buffer.size)\n        val last = buffer.removeLast()\n        val value = if (j < buffer.size) buffer.set(j,
last) else last\n        yield(value)\n    }\n}\n\n/**\n * A sequence that returns the values from the underlying
[sequence] that either match or do not match\n * the specified [predicate].\n *\n * @param sendWhen If `true`,
values for which the predicate returns
`true` are returned. Otherwise,\n * values for which the predicate returns `false` are returned\n */\n\ninternal class
FilteringSequence<T>(\n    private val sequence: Sequence<T>,\n    private val sendWhen: Boolean = true,\n    private val predicate: (T) -> Boolean\n) : Sequence<T> {\n    override fun iterator(): Iterator<T> = object :
Iterator<T> {\n        val iterator = sequence.iterator()\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1
for continue\n        var nextItem: T? = null\n\n        private fun calcNext() {\n            while (iterator.hasNext()) {\n
                val item = iterator.next()\n                if (predicate(item) == sendWhen) {\n                    nextItem = item\n
                    nextState = 1\n                    return\n                }\n            }\n            nextState = 0\n        }\n        override fun
next(): T {\n            if (nextState == -1)\n                calcNext()\n            if (nextState == 0)\n

```

```

        throw NoSuchElementException()\n        val result = nextItem\n        nextItem = null\n        nextState = -1\n    }\n    @Suppress("UNCHECKED_CAST")\n    return result as T\n    }\n    override fun\n    hasNext(): Boolean {\n        if (nextState == -1)\n            calcNext()\n        return nextState == 1\n    }\n}\n\n/**\n * A sequence which returns the results of applying the given [transformer] function to the values\n * in the underlying [sequence].\n */\ninternal class TransformingSequence<T, R>\nconstructor(private val\n    sequence: Sequence<T>, private val transformer: (T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R>\n    = object : Iterator<R> {\n        val iterator = sequence.iterator()\n        override fun next(): R {\n            return\n            transformer(iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\ninternal fun <E>\n    flatten(iterator: (R) -> Iterator<E>): Sequence<E> {\n        return FlatteningSequence<T, R, E>(sequence,\n            transformer, iterator)\n    }\n}\n\n/**\n * A sequence which returns the results of applying the given [transformer]\n * function to the values\n * in the underlying [sequence], where the transformer function takes the index of the value\n * in the underlying\n * sequence along with the value itself.\n */\ninternal class TransformingIndexedSequence<T,\n    R>\nconstructor(private val sequence: Sequence<T>, private val transformer: (Int, T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R> = object : Iterator<R> {\n        val iterator = sequence.iterator()\n        var index = 0\n        override fun next(): R {\n            return transformer(checkIndexOverflow(index++), iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A\n * sequence which combines values from the underlying [sequence] with\n * their indices and returns them as\n * [IndexedValue] objects.\n */\ninternal class\n    IndexingSequence<T>\nconstructor(private val sequence: Sequence<T>) : Sequence<IndexedValue<T>> {\n    override fun iterator(): Iterator<IndexedValue<T>> = object : Iterator<IndexedValue<T>> {\n        val iterator =\n        sequence.iterator()\n        var index = 0\n        override fun next(): IndexedValue<T> {\n            return\n            IndexedValue(checkIndexOverflow(index++), iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence which takes the values from two parallel\n * underlying sequences, passes them to the given\n * [transform] function and returns the values returned by that\n * function. The sequence stops returning\n * values as soon as one of the underlying sequences stops returning\n * values.\n */\ninternal class MergingSequence<T1, T2, V>\nconstructor(\n    private val sequence1:\n    Sequence<T1>,\n    private val\n    sequence2: Sequence<T2>,\n    private val transform: (T1, T2) -> V) : Sequence<V> {\n    override fun iterator():\n    Iterator<V> = object : Iterator<V> {\n        val iterator1 = sequence1.iterator()\n        val iterator2 =\n        sequence2.iterator()\n        override fun next(): V {\n            return transform(iterator1.next(), iterator2.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator1.hasNext() && iterator2.hasNext()\n        }\n    }\n}\n\ninternal class FlatteningSequence<T, R, E>\nconstructor(\n    private val sequence: Sequence<T>,\n    private val transformer: (T) -> R,\n    private val iterator: (R) -> Iterator<E>) : Sequence<E> {\n    override fun\n    iterator(): Iterator<E> = object : Iterator<E> {\n        val iterator = sequence.iterator()\n        var itemIterator:\n        Iterator<E>? = null\n        override fun next(): E {\n            if (!ensureItemIterator())\n                throw\n                NoSuchElementException()\n            return itemIterator!!.next()\n        }\n        override fun hasNext(): Boolean {\n            return ensureItemIterator()\n        }\n        private fun\n        ensureItemIterator(): Boolean {\n            if (itemIterator?.hasNext() == false)\n                itemIterator = null\n            while (itemIterator == null) {\n                if (!iterator.hasNext())\n                    return false\n            } else {\n                val element = iterator.next()\n                val nextItemIterator = iterator(transformer(element))\n            }\n            if (nextItemIterator.hasNext()) {\n                itemIterator = nextItemIterator\n                return true\n            }\n        }\n    }\n}\n\ninternal fun <T, C, R> flatMapIndexed(source:\n    Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> =\n    sequence {\n        var\n        index = 0\n        for (element in source) {\n            val result = transform(checkIndexOverflow(index++), element)\n            yieldAll(iterator(result))\n        }\n    }\n}\n\n/**\n * A sequence that supports drop(n) and take(n) operations\n */\ninternal interface\n    DropTakeSequence<T> : Sequence<T> {\n    fun drop(n: Int): Sequence<T>\n    fun take(n: Int):

```



```

Sequence<T>() {
    /**
     * A sequence that skips [startIndex] values from the underlying [sequence]
     * and stops returning values right before [endIndex], i.e. stops at `endIndex - 1`
     */
    internal class SubSequence<T>() {
        private val sequence: Sequence<T>,
        private val startIndex: Int,
        private val endIndex: Int() : Sequence<T>,
        DropTakeSequence<T> {
            init {
                require(startIndex >= 0) { "\"startIndex should be non-negative, but is $startIndex\"" }
                require(endIndex >= 0) { "\"endIndex should be non-negative, but is $endIndex\"" }
                require(endIndex >= startIndex) { "\"endIndex should be not less than startIndex, but was $endIndex < $startIndex\"" }
            }
            private val count: Int get() = endIndex - startIndex
            override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)
            override fun take(n: Int): Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)
            override fun iterator(): Iterator<T> = object : Iterator<T> {
                val iterator = sequence.iterator()
                var position = 0 // Shouldn't be called from constructor to avoid premature iteration
                private fun drop() { while (position < startIndex && iterator.hasNext()) { iterator.next(); position++; } }
                override fun hasNext(): Boolean { drop(); return (position < endIndex) && iterator.hasNext() }
                override fun next(): T { drop(); if (position >= endIndex) throw NoSuchElementException()
                    position++; return iterator.next() }
            }
        }
    }
}
/**
 * A sequence that returns at most [count] values from the underlying [sequence], and stops returning values
 * as soon as that count is reached.
 */
internal class TakeSequence<T>() {
    private val sequence: Sequence<T>,
    private val count: Int() : Sequence<T>,
    DropTakeSequence<T> {
        init {
            require(count >= 0) { "\"count must be non-negative, but was $count.\""}
        }
        override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, n, count)
        override fun take(n: Int): Sequence<T> = if (n >= count) this else TakeSequence(sequence, n)
        override fun iterator(): Iterator<T> = object : Iterator<T> {
            var left = count
            val iterator = sequence.iterator()
            override fun next(): T {
                if (left == 0) throw NoSuchElementException()
                left--
                return iterator.next()
            }
            override fun hasNext(): Boolean { return left > 0 && iterator.hasNext() }
        }
    }
}
/**
 * A sequence that returns values from the underlying [sequence] while the [predicate] function returns `true`, and stops returning values once the function returns `false` for the next element.
 */
internal class TakeWhileSequence<T>() {
    constructor(
        private val sequence: Sequence<T>,
        private val predicate: (T) -> Boolean() : Sequence<T> {
            override fun iterator(): Iterator<T> = object : Iterator<T> {
                val iterator = sequence.iterator()
                var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue
                var nextItem: T? = null
                private fun calcNext() {
                    if (iterator.hasNext()) {
                        val item = iterator.next()
                        if (predicate(item)) {
                            nextState = 1
                            nextItem = item
                            return
                        }
                        nextState = 0
                    }
                }
                override fun next(): T {
                    if (nextState == -1) calcNext() // will change nextState
                    if (nextState == 0) throw NoSuchElementException()
                    @Suppress("UNCHECKED_CAST") val result = nextItem as T // Clean next to avoid keeping reference on yielded instance
                    nextItem = null
                    nextState = -1
                    return result
                }
                override fun hasNext(): Boolean {
                    if (nextState == -1) calcNext() // will change nextState
                    return nextState == 1
                }
            }
        }
    }
}
/**
 * A sequence that skips the specified number of values from the underlying [sequence] and returns
 * all values after that.
 */
internal class DropSequence<T>() {
    private val sequence: Sequence<T>,
    private val count: Int() : Sequence<T>, DropTakeSequence<T> {
        init {
            require(count >= 0) { "\"count must be non-negative, but was $count.\""}
        }
        override fun drop(n: Int): Sequence<T> = (count + n).let { n1 -> if (n1 < 0) DropSequence(this, n) else DropSequence(sequence, n1) }
        override fun take(n: Int): Sequence<T> = (count + n).let { n1 -> if (n1 < 0) TakeSequence(this, n) else SubSequence(sequence, count, n1) }
        override fun iterator(): Iterator<T> = object : Iterator<T> {
            val iterator = sequence.iterator()
            var left = count // Shouldn't be called from constructor to avoid premature iteration
            private fun drop() { while (left > 0 && iterator.hasNext()) { iterator.next(); left--; } }
            override fun next(): T { drop(); return iterator.next() }
        }
    }
}

```

```

iterator.next()\n    }\n\n    override fun hasNext(): Boolean {\n        drop()\n        return iterator.hasNext()\n    }\n}\n\n/**\n * A sequence
that skips the values from the underlying [sequence] while the given [predicate] returns `true` and returns\n * all
values after that.\n */\ninternal class DropWhileSequence<T>\nconstructor(\n    private val sequence:
Sequence<T>,\n    private val predicate: (T) -> Boolean\n) : Sequence<T> {\n\n    override fun iterator():
Iterator<T> = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var dropState: Int = -1 // -1 for not
dropping, 1 for nextItem, 0 for normal iteration\n        var nextItem: T? = null\n        private fun drop() {\n
while (iterator.hasNext()) {\n            val item = iterator.next()\n            if (!predicate(item)) {\n
nextItem = item\n                dropState = 1\n                return\n            }\n            dropState = 0\n
}\n\n        override fun next(): T {\n            if (dropState == -1)\n                drop()\n            if (dropState == 1)
{\n                @Suppress("UNCHECKED_CAST")\n                val result = nextItem as T\n                nextItem =
null\n                dropState = 0\n                return result\n            }\n            return iterator.next()\n        }\n\n
override fun hasNext(): Boolean {\n            if (dropState == -1)\n                drop()\n            return dropState == 1 ||
iterator.hasNext()\n        }\n    }\n}\n\ninternal class DistinctSequence<T, K>(private val source: Sequence<T>,\n    private val keySelector: (T) -> K) : Sequence<T> {\n    override fun iterator(): Iterator<T> =
DistinctIterator(source.iterator(), keySelector)\n}\n\nprivate class DistinctIterator<T, K>(private val source:
Iterator<T>,\n    private val keySelector: (T) -> K) : AbstractIterator<T>() {\n    private val observed =
HashSet<K>()\n    override fun computeNext() {\n        while (source.hasNext()) {\n            val next =
source.next()\n            val key = keySelector(next)\n            if (observed.add(key))\n                {\n                    setNext(next)\n                    return\n                }\n            done()\n        }\n    }\n}\n\nprivate class
GeneratorSequence<T : Any>(private val getInitialValue: () -> T?, private val getNextValue: (T) -> T?) :
Sequence<T> {\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        var nextItem: T? = null\n
var nextState: Int = -2 // -2 for initial unknown, -1 for next unknown, 0 for done, 1 for continue\n        private fun
calcNext() {\n            nextItem = if (nextState == -2) getInitialValue() else getNextValue(nextItem!!)\n
nextState = if (nextItem == null) 0 else 1\n        }\n\n        override fun next(): T {\n            if (nextState < 0)\n                calcNext()\n            if (nextState == 0)\n                throw NoSuchElementException()\n            val result =
nextItem as T\n            // Do not clean nextItem (to avoid keeping reference on yielded instance) -- need to keep
state for\n            getNextValue\n            nextState = -1\n            return result\n        }\n\n        override fun hasNext(): Boolean {\n
            if (nextState < 0)\n                calcNext()\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * Returns a
wrapper sequence that provides values of this sequence, but ensures it can be iterated only one time.\n */\n * The
operation is _intermediate_ and _stateless_.\n */\n * [IllegalStateException] is thrown on iterating the returned
sequence for the second time and the following times.\n */\n * ^\npublic fun <T> Sequence<T>.constrainOnce():
Sequence<T> {\n    // as? does not work in js\n    //return this as? ConstrainedOnceSequence<T> ?:\n
ConstrainedOnceSequence(this)\n    return if (this is ConstrainedOnceSequence<T>) this else
ConstrainedOnceSequence(this)\n}\n\n/**\n * Returns a sequence which invokes the function to calculate the next
value on each iteration until the function returns `null`.\n */\n * The returned sequence is constrained
to be iterated only once.\n */\n * @see constrainOnce\n * @see kotlin.sequences.sequence\n */\n * @sample
samples.collections.Sequences.Building.generateSequence\n */\npublic fun <T : Any>
generateSequence(nextFunction: () -> T?): Sequence<T> {\n    return GeneratorSequence(nextFunction, {\n
nextFunction() }).constrainOnce()\n}\n\n/**\n * Returns a sequence defined by the starting value [seed] and the
function [nextFunction],\n */\n * which is invoked to calculate the next value based on the previous one on each
iteration.\n */\n * The sequence produces values until it encounters first `null` value.\n */\n * If [seed] is `null`, an empty
sequence is produced.\n */\n * The sequence can be iterated multiple times, each time starting with [seed].\n */\n *
@see kotlin.sequences.sequence\n */\n * @sample
samples.collections.Sequences.Building.generateSequenceWithSeed\n
*/\n@kotlin.internal.LowPriorityInOverloadResolution\npublic fun <T : Any> generateSequence(seed: T?,
nextFunction: (T) -> T?): Sequence<T>

```



```

IllegalStateException(message.toString())\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and
 * Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
 * can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS
 * AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
 * https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.js.*\nimport
 * primitiveArrayConcat\nimport withType\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n *
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds
 * of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun <T>
 * Array<out T>.elementAt(index: Int): T {\n    return elementAtOrElse(index) { throw
 * IndexOutOfBoundsException("index: $index, size: $size}") }\n}\n\n/**\n * Returns an element at the given
 * [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n
 * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun
 * ByteArray.elementAt(index: Int): Byte {\n    return elementAtOrElse(index) { throw
 * IndexOutOfBoundsException("index: $index, size: $size}") }\n}\n\n/**\n * Returns an element at the given
 * [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
 * samples.collections.Collections.Elements.elementAt\n */\npublic actual fun ShortArray.elementAt(index: Int): Short
 * {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")
 * }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
 * out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
 * actual fun IntArray.elementAt(index: Int): Int {\n    return elementAtOrElse(index) { throw
 * IndexOutOfBoundsException("index: $index, size: $size") }\n}\n\n/**\n
 * \n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of
 * bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun
 * LongArray.elementAt(index: Int): Long {\n    return elementAtOrElse(index) { throw
 * IndexOutOfBoundsException("index: $index, size: $size") }\n}\n\n/**\n * Returns an element at the given
 * [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
 * samples.collections.Collections.Elements.elementAt\n */\npublic actual fun FloatArray.elementAt(index: Int): Float
 * {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")
 * }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
 * out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
 * actual fun DoubleArray.elementAt(index:
 * Int): Double {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size:
 * $size") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
 * [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
 * \n */\npublic actual fun BooleanArray.elementAt(index: Int): Boolean {\n    return elementAtOrElse(index) { throw
 * IndexOutOfBoundsException("index: $index, size: $size") }\n}\n\n/**\n * Returns an element at the given
 * [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
 * samples.collections.Collections.Elements.elementAt\n */\npublic actual fun CharArray.elementAt(index: Int): Char
 * {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")
 * }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic actual fun <T> Array<out T>.asList():
 * List<T> {\n    return ArrayList<T>(this.unsafeCast<Array<Any?>>())\n}\n\n/**\n * Returns a [List] that wraps the
 * original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun ByteArray.asList(): List<Byte> {\n
 * return this.unsafeCast<Array<Byte>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
 * \n */\n@kotlin.internal.InlineOnly\npublic actual inline fun ShortArray.asList(): List<Short> {\n    return
 * this.unsafeCast<Array<Short>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
 * \n */\n@kotlin.internal.InlineOnly\npublic actual inline fun IntArray.asList(): List<Int> {\n    return
 * this.unsafeCast<Array<Int>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
 * \n */\n@kotlin.internal.InlineOnly\npublic actual inline fun LongArray.asList(): List<Long> {\n    return

```

```

this.unsafeCast<Array<Long>>().asList()\n\n\n**\n * Returns a [List] that wraps the original array.\n
*\n@kotlin.internal.InlineOnly\npublic
  actual inline fun FloatArray.asList(): List<Float> {\n  return this.unsafeCast<Array<Float>>().asList()\n}\n\n\n**\n
* Returns a [List] that wraps the original array.\n *\n@kotlin.internal.InlineOnly\npublic actual inline fun
DoubleArray.asList(): List<Double> {\n  return this.unsafeCast<Array<Double>>().asList()\n}\n\n\n**\n
* Returns
a [List] that wraps the original array.\n *\n@kotlin.internal.InlineOnly\npublic actual inline fun
BooleanArray.asList(): List<Boolean> {\n  return this.unsafeCast<Array<Boolean>>().asList()\n}\n\n\n**\n
*
Returns a [List] that wraps the original array.\n *\npublic actual fun CharArray.asList(): List<Char> {\n  return
object : AbstractList<Char>(), RandomAccess {\n    override val size: Int get() = this@asList.size\n    override
fun isEmpty(): Boolean = this@asList.isEmpty()\n    override fun contains(element: Char): Boolean =
this@asList.contains(element)\n    override fun get(index: Int): Char {\n
AbstractList.checkElementIndex(index,
size)\n    return this@asList[index]\n    }\n    override fun indexOf(element: Char): Int {\n
@Suppress(\\"USELESS_CAST\")\n    if ((element as Any?) !is Char) return -1\n    return
this@asList.indexOf(element)\n    }\n    override fun lastIndexOf(element: Char): Int {\n
@Suppress(\\"USELESS_CAST\")\n    if ((element as Any?) !is Char) return -1\n    return
this@asList.lastIndexOf(element)\n    }\n  }\n}\n\n\n**\n
* Returns `true` if the two specified arrays are *deeply*
equal to one another.\n * i.e. contain the same number of the same elements in the same order.\n * \n * If two
corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any
nesting level the behavior is undefined.\n * \n * The elements of other types are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN`
is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin(\\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual infix fun <T>
Array<out T>.contentDeepEquals(other: Array<out T>): Boolean {\n  return
this.contentDeepEquals(other)\n}\n\n\n**\n
* Returns `true` if the two specified arrays are *deeply* equal to one
another.\n * i.e. contain the same number of the same elements in the same order.\n * \n * The specified arrays are
also considered deeply equal if both are `null`.\n * \n * If two corresponding elements are nested arrays, they are
also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n * The
elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point
numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin(\\"1.4\")\n@library(\\"arrayDeepEquals\")\npublic actual infix fun <T> Array<out
T>?.contentDeepEquals(other:
Array<out T>?): Boolean {\n  definedExternally\n}\n\n\n**\n
* Returns a hash code based on the contents of this
array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting
level the behavior is undefined.\n
*\n@SinceKotlin(\\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out
T>.contentDeepHashCode(): Int {\n  return this.contentDeepHashCode()\n}\n\n\n**\n
* Returns a hash code based
on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays
contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin(\\"1.4\")\n@library(\\"arrayDeepHashCode\")\npublic actual fun <T> Array<out
T>?.contentDeepHashCode(): Int {\n  definedExternally\n}\n\n\n**\n
* Returns a string representation of the
contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n
* If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent
recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
*\n@SinceKotlin(\\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out
T>.contentDeepToString(): String {\n  return this.contentDeepToString()\n}\n\n\n**\n
* Returns a string
representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n
* If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n

```

```

@sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
*\n@SinceKotlin("1.4")\n@library("arrayDeepToString")\npublic actual fun <T> Array<out
T>?.contentDeepToString(): String {\n    definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays
are *structurally* equal to one another,\n
 * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for
equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself
and `-0.0` is not equal to `0.0`.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun <T>
Array<out T>.contentEquals(other: Array<out T>): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince
= "1.4")\npublic actual infix fun ByteArray.contentEquals(other: ByteArray): Boolean {\n    return
this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one
another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are
compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is
equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
ShortArray.contentEquals(other: ShortArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for
equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself
and `-0.0` is not equal to `0.0`.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
IntArray.contentEquals(other: IntArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true`
if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same
elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
LongArray.contentEquals(other: LongArray): Boolean
{\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally*
equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements
are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that
`NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid
deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual
infix fun FloatArray.contentEquals(other: FloatArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and
\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and
`-0.0` is not equal to `0.0`.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
DoubleArray.contentEquals(other: DoubleArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and
`-0.0` is not equal to `0.0`.\n

```

```

*\/n@Deprecated(\`Use Kotlin compiler 1.4 to avoid deprecation
warning.\`)\n@SinceKotlin(\`1.1\`)\n@DeprecatedSinceKotlin(hiddenSince = \`1.4\`)\npublic actual infix fun
BooleanArray.contentEquals(other: BooleanArray): Boolean {\n  return this.contentEquals(other)\n}\n\n**\n *
Returns `true` if the two specified arrays are *structurally*
equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The
elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means
that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\/n@Deprecated(\`Use Kotlin compiler 1.4 to avoid
deprecation warning.\`)\n@SinceKotlin(\`1.1\`)\n@DeprecatedSinceKotlin(hiddenSince = \`1.4\`)\npublic actual
infix fun CharArray.contentEquals(other: CharArray): Boolean {\n  return this.contentEquals(other)\n}\n\n**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun
<T> Array<out T>?.contentEquals(other: Array<out T>?): Boolean {\n  definedExternally\n}\n\n**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun ByteArray?.contentEquals(other:
ByteArray?): Boolean {\n  definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic
actual infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean {\n  definedExternally\n}\n\n**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun IntArray?.contentEquals(other:
IntArray?): Boolean {\n  definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal
to `0.0`.\n *\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun
LongArray?.contentEquals(other: LongArray?): Boolean {\n  definedExternally\n}\n\n**\n * Returns `true` if the
two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements
in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For
floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun FloatArray?.contentEquals(other:
FloatArray?): Boolean {\n  definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN`
is equal to itself and `-0.0` is not equal to `0.0`.\n *\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic
actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {\n  definedExternally\n}\n\n**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\/n@SinceKotlin(\`1.4\`)\n@library(\`arrayEquals\`)\npublic actual infix fun BooleanArray?.contentEquals(other:

```

BooleanArray?): Boolean {
 definedExternally
 Returns `true` if the two specified arrays are
 structurally equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 @SinceKotlin("1.4")
 library("arrayEquals")
 public actual infix fun CharArray?.contentEquals(other:
 CharArray?): Boolean {
 definedExternally
 Returns a hash code based on the contents of this array
 as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun <T>
 Array<out T>.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based
 on the contents of this array as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 ByteArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as
 if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 ShortArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 IntArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 LongArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as if it is [List].
 @Deprecated("Use
 Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 FloatArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 DoubleArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based
 on the contents of this array as if it is [List].
 @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 BooleanArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based
 on the contents of this array as if it is [List].
 @Deprecated("Use
 Kotlin compiler 1.4 to avoid deprecation
 warning.")
 @SinceKotlin("1.1")
 @DeprecatedSinceKotlin(hiddenSince = "1.4")
 public actual fun
 CharArray.contentHashCode(): Int {
 return this.contentHashCode()
 Returns a hash code based on
 the contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public actual
 fun <T> Array<out T>?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on
 the contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public actual
 fun ByteArray?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on the
 contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public actual fun
 ShortArray?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on the
 contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public
 actual fun IntArray?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on
 the contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public actual
 fun LongArray?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on the
 contents of this array as if it is [List].
 @SinceKotlin("1.4")
 library("arrayHashCode")
 public actual fun
 FloatArray?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on the


```

contents of this array as if it is [List].\n *\/\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
DoubleArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n *\/\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
BooleanArray?.contentHashCode(): Int
{\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\/\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun CharArray?.contentHashCode(): Int
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun <T>
Array<out T>.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince
= "1.4")\npublic actual fun ByteArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n *
Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun ShortArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun IntArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the
contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun LongArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun FloatArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun
DoubleArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun BooleanArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun CharArray.contentToString():
String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the
specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun <T> Array<out T>?.contentToString():

```

```

String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ByteArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ShortArray?.contentToString():
String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun IntArray?.contentToString(): String {\n
definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun LongArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun FloatArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun DoubleArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun BooleanArray?.contentToString():
String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun CharArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Copies this array or
its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the
[destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param
destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by
default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param
endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the
[destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic
actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0,
endIndex: Int = size): Array<T> {\n  arrayCopy(this, destination, destinationOffset, startIndex, endIndex)\n
return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n
*\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with
the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in
the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to
copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified
[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the

```

```

[destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): ByteArray {\n    arrayCopy(this.unsafeCast<Array<Byte>>(),
destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n**\n* Copies this array or its subrange into the [destination] array and returns that array.\n* \n* It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination
range.\n* \n* @param destination the array to copy to.\n* @param destinationOffset the position in the
[destination] array to copy to, 0 by default.\n* @param startIndex the beginning (inclusive) of the subrange to copy,
0 by default.\n* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n* \n*
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of this array indices or when `startIndex > endIndex`.\n* @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n* or when that index is
out of the [destination] array indices range.\n* \n* @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int =
0, startIndex: Int =
0, endIndex: Int = size): ShortArray {\n    arrayCopy(this.unsafeCast<Array<Short>>(),
destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n**\n* Copies this array or its subrange into the [destination] array and returns that array.\n* \n* It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n* \n*
@param destination the array to copy to.\n* @param destinationOffset the position in the [destination] array to
copy to, 0 by default.\n* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n*
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n* \n* @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n* @throws IndexOutOfBoundsException when the subrange
doesn't
fit into the [destination] array starting at the specified [destinationOffset],\n* or when that index is out of the
[destination] array indices range.\n* \n* @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): IntArray {\n    arrayCopy(this.unsafeCast<Array<Int>>(),
destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n**\n* Copies this array or its subrange into the [destination] array and returns that array.\n* \n* It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n* \n*
@param destination the array to copy to.\n* @param destinationOffset the position in the [destination] array to
copy to, 0 by default.\n
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n* @param endIndex the end
(exclusive) of the subrange to copy, size of this array by default.\n* \n* @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n* or when that index is out of the [destination] array indices range.\n*
\n* @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): LongArray {\n    arrayCopy(this.unsafeCast<Array<Long>>(),
destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex,

```

endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray {\n    arrayCopy(this.unsafeCast<Array<Float>>(), destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {\n    arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic\nactual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {\n    arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or
```

[endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
 or when that index is out of the [destination] array indices range.
 @return the [destination] array.

```

@SinceKotlin("1.3")@kotlin.internal.InlineOnly@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): CharArray {
    arrayCopy(this.unsafeCast<Array<Char>>(),
destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)
    return destination
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline fun <T> Array<out T>.copyOf(): Array<T> {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("NOTHING_TO_INLINE")
public actual inline fun ByteArray.copyOfOf(): ByteArray {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("NOTHING_TO_INLINE")
public actual inline fun ShortArray.copyOfOf(): ShortArray {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("NOTHING_TO_INLINE")
public actual inline fun IntArray.copyOfOf(): IntArray {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

public actual fun LongArray.copyOfOf(): LongArray {
    return withType("LongArray", this.asDynamic().slice())
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("NOTHING_TO_INLINE")
public actual inline fun FloatArray.copyOfOf(): FloatArray {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

@Suppress("NOTHING_TO_INLINE")
public actual inline fun DoubleArray.copyOfOf(): DoubleArray {
    return this.asDynamic().slice()
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

public actual fun BooleanArray.copyOfOf(): BooleanArray {
    return withType("BooleanArray", this.asDynamic().slice())
}

Returns new array which is a copy of the original array.
@sample
samples.collections.Arrays.CopyOfOperations.copyOfOf

public actual fun CharArray.copyOfOf(newSize: Int): CharArray {
    return withType("CharArray", this.asDynamic().slice(), newSize)
}

Returns new array which is a copy of the original array, resized to the given [newSize].
The copy is either truncated or padded at the end with zero values if necessary.
- If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
- If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

public actual fun ByteArray.copyOfOf(newSize: Int): ByteArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, ByteArray(newSize))
}

Returns new array which is a copy of the original array, resized to the given [newSize].
The copy is either truncated or padded at the end with zero values if necessary.
- If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
- If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

public actual fun ShortArray.copyOfOf(newSize: Int): ShortArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, ShortArray(newSize))
}

Returns new array which is a copy of the original

```

array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
IntArray.copyOf(newSize: Int): IntArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, IntArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
LongArray.copyOf(newSize: Int): LongArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"LongArray\", arrayCopyResize(this, newSize, 0L))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
FloatArray.copyOf(newSize: Int): FloatArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, FloatArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
DoubleArray.copyOf(newSize: Int): DoubleArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, DoubleArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `false` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
BooleanArray.copyOf(newSize: Int): BooleanArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"BooleanArray\", arrayCopyResize(this, newSize, false))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
CharArray.copyOf(newSize: Int): CharArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"CharArray\", fillFrom(this, CharArray(newSize)))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `null` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n *\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return arrayCopyResize(this,

@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n\npublic actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType("BooleanArray",\n
 this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which\n
 is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws\n
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n\npublic actual fun\n
 CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType("CharArray",\n
 this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the specified\n
 [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param\n
 toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws\n
 IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size\n
 of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 \n\n * \n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic\n
 actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n this.asDynamic().fill(element, fromIndex,\n
 toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param\n
 fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)\n
 to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero\n
 or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater\n
 than [toIndex].\n
 \n\n * \n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic\n
 actual fun ByteArray.fill(element: Byte, fromIndex:\n
 Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n
 this.asDynamic().fill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified\n
 [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param\n
 toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws\n
 IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n
 \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 \n\n * \n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic\n
 actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n this.asDynamic().fill(element, fromIndex,\n
 toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified\n
 [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param\n
 toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws\n
 IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n
 \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 \n\n * \n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic\n
 actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n this.asDynamic().fill(element, fromIndex,\n
 toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param\n
 fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)\n
 to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException


```

if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*/\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*/\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the
range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to
fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n
*/\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n
*/\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline operator
fun <T> Array<out T>.plus(element: T): Array<T> {\n    return
this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original
array and then the given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline operator fun ByteArray.plus(element: Byte): ByteArray {\n    return
plus(byteArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the
given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
ShortArray.plus(element: Short): ShortArray {\n    return plus(shortArrayOf(element))\n}\n\n/**\n * Returns an
array containing all elements of the original array and then the given [element].\n

```

```

*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(element: Int):
IntArray {\n    return plus(intArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun LongArray.plus(element: Long): LongArray {\n    return plus(longArrayOf(element))\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given
[element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
FloatArray.plus(element: Float): FloatArray {\n    return plus(floatArrayOf(element))\n}\n\n/**\n * Returns an array
containing all elements of the original array and then the given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
DoubleArray.plus(element:
Double): DoubleArray {\n    return plus(doubleArrayOf(element))\n}\n\n/**\n * Returns an array containing all
elements of the original array and then the given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {\n    return
plus(booleanArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then
the given [element].\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
CharArray.plus(element: Char): CharArray {\n    return plus(charArrayOf(element))\n}\n\n/**\n * Returns
an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual operator fun <T> Array<out T>.plus(elements:
Collection<T>): Array<T> {\n    return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic
actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {\n    return
fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual
operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {\n    return
fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and
then all elements of the given [elements] collection.\n */\npublic actual operator fun IntArray.plus(elements:
Collection<Int>): IntArray {\n    return fillFromCollection(this.copyOf(size + elements.size), this.size,
elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the
given [elements] collection.\n */\npublic actual operator fun LongArray.plus(elements: Collection<Long>):
LongArray {\n    return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of
the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun
FloatArray.plus(elements: Collection<Float>): FloatArray {\n    return fillFromCollection(this.copyOf(size +
elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and
then all elements of the given [elements] collection.\n */\npublic actual operator fun DoubleArray.plus(elements:
Collection<Double>): DoubleArray {\n    return fillFromCollection(this.copyOf(size + elements.size), this.size,
elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the
given [elements] collection.\n */\npublic actual operator fun BooleanArray.plus(elements: Collection<Boolean>):
BooleanArray {\n    return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all
elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator
fun CharArray.plus(elements: Collection<Char>): CharArray {\n    return fillFromCollection(this.copyOf(size +
elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and
then all elements of the given [elements] array.\n */\n@Suppress("ACTUAL_WITHOUT_EXPECT",
"NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>):
Array<T> {\n    return this.asDynamic().concat(elements)\n}\n\n/**\n * Returns an array containing all elements of
the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements:
ByteArray): ByteArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] array.\n */

```

```

*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(elements:
ShortArray): ShortArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray):
IntArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements:
LongArray): LongArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements:
FloatArray): FloatArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements:
DoubleArray): DoubleArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements]
array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
BooleanArray.plus(elements: BooleanArray): BooleanArray {\n    return primitiveArrayConcat(this,
elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the
given [elements] array.\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
CharArray.plus(elements: CharArray): CharArray {\n    return primitiveArrayConcat(this, elements)\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline fun <T>
Array<out T>.plusElement(element: T): Array<T> {\n    return
this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun
IntArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n
* Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\npublic actual fun
LongArray.sort(): Unit {\n    @Suppress("DEPRECATION")\n    if (size > 1) sort { a: Long, b: Long ->
a.compareTo(b) }\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun
ByteArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun
ShortArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun
DoubleArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic
actual fun FloatArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun
CharArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place according to the natural order
of its elements.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other
after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n
*\npublic actual fun
<T : Comparable<T>> Array<out T>.sort(): Unit {\n    if (size > 1) sortArray(this)\n}\n\n/**\n * Sorts the array in-
place according to the order specified by the given [comparison] function.\n * \n * The sort is _stable_. It means that
equal elements preserve their order relative to each other after sorting.\n
*\n@Deprecated("Use sortWith instead",
ReplaceWith("this.sortWith(Comparator(comparison))"))\n@DeprecatedSinceKotlin(warningSince
= "1.6")\npublic fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit {\n    if (size > 1)
sortArrayWith(this, comparison)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the

```

```

start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size):
Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArrayWith(this, fromIndex, toIndex,
naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n  subarray.sort()\n}\n\n/**\n * Sorts a
range
in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param
toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n  subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive)
to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<IntArray>()\n  subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size
of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArrayWith(this.unsafeCast<Array<Long>>(),
fromIndex, toIndex, naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex
the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort,
size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n

```

```

*^@SinceKotlin("1.4")^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^npublic
actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {^n
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)^n    val subarray =
    this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<FloatArray>()^n    subarray.sort()^n}^n/n/**^n * Sorts a
range in the array in-place.^n * ^n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.^n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.^n * ^n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.^n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].^n * ^n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray^n
*^@SinceKotlin("1.4")^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^npublic
actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {^n
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)^n    val subarray =
    this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<DoubleArray>()^n    subarray.sort()^n}^n/n/**^n * Sorts
a range in the array in-place.^n * ^n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.^n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.^n * ^n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.^n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].^n * ^n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray^n
*^@SinceKotlin("1.4")^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^npublic
actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {^n
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)^n    val subarray =
    this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()^n    subarray.sort()^n}^n/n/**^n
    * Sorts the array in-place according to the order specified by the given [comparison] function.^n
*^@Deprecated("Use other sorting functions from the Standard
Library")^n@DeprecatedSinceKotlin(warningSince = "1.6")^n@kotlin.internal.InlineOnly^npublic inline fun
ByteArray.sort(noinline comparison: (a: Byte, b: Byte) -> Int): Unit {^n    asDynamic().sort(comparison)^n}^n/n/**^n
    * Sorts the array in-place according to the order specified by the given [comparison] function.^n
*^@Deprecated("Use other sorting functions from the Standard
Library")^n@DeprecatedSinceKotlin(warningSince = "1.6")^n@kotlin.internal.InlineOnly^npublic inline fun
ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {^n
    asDynamic().sort(comparison)^n}^n/n/n/**^n * Sorts the array in-place according to the order specified by the given
[comparison] function.^n *^@Deprecated("Use other sorting functions from the Standard
Library")^n@DeprecatedSinceKotlin(warningSince
= "1.6")^n@kotlin.internal.InlineOnly^npublic inline fun IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int):
Unit {^n    asDynamic().sort(comparison)^n}^n/n/n/**^n * Sorts the array in-place according to the order specified by
the given [comparison] function.^n *^@Deprecated("Use other sorting functions from the Standard
Library")^n@DeprecatedSinceKotlin(warningSince = "1.6")^n@kotlin.internal.InlineOnly^npublic inline fun
LongArray.sort(noinline comparison: (a: Long, b: Long) -> Int): Unit {^n
    asDynamic().sort(comparison)^n}^n/n/n/**^n * Sorts the array in-place according to the order specified by the given
[comparison] function.^n *^@Deprecated("Use other sorting functions from the Standard
Library")^n@DeprecatedSinceKotlin(warningSince = "1.6")^n@kotlin.internal.InlineOnly^npublic inline fun
FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {^n
    asDynamic().sort(comparison)^n}^n/n/n/**^n * Sorts the array in-place according to the
order specified by the given [comparison] function.^n *^@Deprecated("Use other sorting functions from the
Standard Library")^n@DeprecatedSinceKotlin(warningSince = "1.6")^n@kotlin.internal.InlineOnly^npublic inline
fun DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {^n
    asDynamic().sort(comparison)^n}^n/n/n/**^n * Sorts the array in-place according to the order specified by the given
[comparison] function.^n *^@Deprecated("Use other sorting functions from the Standard

```



```

K> Comparator<T>.thenBy(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n
Comparator { a, b ->\n    val previousCompare = this@thenBy.compare(a, b)\n    if (previousCompare != 0)
previousCompare else compareValuesBy(a, b, comparator, selector)\n } \n\n/**\n * Creates a descending
comparator using the primary comparator and\n * the function to transform value to a [Comparable] instance for
comparison.\n *\n * @sample samples.comparisons.Comparisons.thenByDescending\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Comparator<T>.thenByDescending(crossinline selector: (T)
-> Comparable<*>?): Comparator<T> =\n    Comparator { a, b ->\n        val previousCompare =
this@thenByDescending.compare(a, b)\n        if (previousCompare != 0) previousCompare else
compareValuesBy(b,
a, selector)\n    } \n\n/**\n * Creates a descending comparator comparing values after the primary comparator
defined them equal. It uses\n * the [selector] function to transform values and then compares them with the given
[comparator].\n *\n * @sample samples.comparisons.Comparisons.thenByDescendingWithComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> Comparator<T>.thenByDescending(comparator:
Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n    Comparator { a, b ->\n        val
previousCompare = this@thenByDescending.compare(a, b)\n        if (previousCompare != 0) previousCompare else
compareValuesBy(b, a, comparator, selector)\n    } \n\n/**\n * Creates a comparator using the primary comparator
and function to calculate a result of comparison.\n *\n * @sample
samples.comparisons.Comparisons.thenComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int):
Comparator<T> =\n    Comparator { a, b ->\n        val previousCompare = this@thenComparator.compare(a, b)\n
        if (previousCompare != 0) previousCompare else comparison(a, b)\n    } \n\n/**\n * Combines this comparator and
the given [comparator] such that the latter is applied only\n * when the former considered values equal.\n *\n *
@sample samples.comparisons.Comparisons.then\n
*\npublic infix fun <T> Comparator<T>.then(comparator:
Comparator<in T>): Comparator<T> =\n    Comparator { a, b ->\n        val previousCompare =
this@then.compare(a, b)\n        if (previousCompare != 0) previousCompare else comparator.compare(a, b)\n
    } \n\n/**\n * Combines this comparator and the given [comparator] such that the latter is applied only\n * when the
former considered values equal.\n *\n * @sample samples.comparisons.Comparisons.thenDescending\n
*\npublic infix fun <T> Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =\n
Comparator<T> { a, b ->\n    val previousCompare = this@thenDescending.compare(a, b)\n    if (previousCompare != 0) previousCompare
else comparator.compare(b, a)\n } \n\n// Not so useful without type inference for receiver of expression\n\n/**\n *
Extends the given [comparator] of non-nullable values to a comparator of nullable values\n * considering `null`
value less than any other value.\n *\n * @sample
samples.comparisons.Comparisons.nullsFirstLastWithComparator\n
*\npublic fun <T : Any>
nullsFirst(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a
=== b -> 0\n            a == null -> -1\n            b == null -> 1\n            else -> comparator.compare(a, b)\n        }\n
    } \n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value less than any other
value.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsFirst():
Comparator<T?> = nullsFirst(naturalOrder())\n\n/**\n * Extends the given [comparator] of non-nullable values to a
comparator of nullable values\n * considering `null` value greater than any other value.\n *\n * @sample
samples.comparisons.Comparisons.nullsFirstLastWithComparator\n
*\npublic fun <T : Any>
nullsLast(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a
=== b -> 0\n            a == null -> 1\n            b == null -> -1\n            else -> comparator.compare(a, b)\n        }\n
    } \n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value greater than any
other value.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> =
nullsLast(naturalOrder())\n\n/**\n * Returns a comparator that compares [Comparable] objects in natural order.\n

```



```

*\n * @sample samples.comparisons.Comparisons.naturalOrderComparator\n
*\npublic fun <T : Comparable<T>> naturalOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")
(NaturalOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that compares [Comparable] objects
in reversed natural order.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n
*\npublic fun <T : Comparable<T>> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")
(ReverseOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that imposes the reverse ordering
of this comparator.\n *\n * @sample samples.comparisons.Comparisons.reversed\n
*\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic fun <T> Comparator<T>.reversed():
Comparator<T> = when (this) {\n    is ReversedComparator -> this.comparator\n    NaturalOrderComparator ->
@Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)\n
ReverseOrderComparator -> @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as
Comparator<T>)\n
    else -> ReversedComparator(this)\n}\n\nprivate class ReversedComparator<T>(public val comparator:
Comparator<T>) : Comparator<T> {\n    override fun compare(a: T, b: T): Int = comparator.compare(b, a)\n
@Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<T> = comparator\n}\n\nprivate
object NaturalOrderComparator : Comparator<Comparable<Any>> {\n    override fun compare(a:
Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)\n
@Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<Comparable<Any>> =
ReverseOrderComparator\n}\n\nprivate object ReverseOrderComparator : Comparator<Comparable<Any>> {\n
override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)\n
@Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<Comparable<Any>> =
NaturalOrderComparator\n}\n\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StandardKt")\npackage kotlin\nimport
kotlin.contracts.*\n\n/**\n * An exception is thrown to indicate that a method body remains to be implemented.\n
*\npublic class NotImplementedError(message: String = "An operation is not implemented.") :
Error(message)\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n
*\n\n@kotlin.internal.InlineOnly\npublic inline fun TODO(): Nothing = throw NotImplementedError()\n\n/**\n *
Always throws [NotImplementedError] stating that operation is not implemented.\n *\n * @param reason a string
explaining why the implementation is missing.\n *\n@kotlin.internal.InlineOnly\npublic inline fun TODO(reason:
String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")\n\n\n/**\n * Calls
the specified function [block] and returns its result.\n
*\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <R> run(block: () -> R): R {\n    contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n    return block()\n}\n\n/**\n * Calls the specified
function [block] with `this` value as its receiver and returns its result.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.run(block: T.() -> R): R {\n    contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n    return block()\n}\n\n/**\n * Calls the specified
function [block] with the given [receiver] as its receiver and returns its result.\n *\n * For detailed usage information
see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#with).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> with(receiver: T, block: T.() -> R): R {\n    contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n    return receiver.block()\n}\n\n/**\n * Calls the
specified function [block] with `this` value as its receiver and returns `this` value.\n *\n * For detailed usage
information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-
functions.html#apply).\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> T.apply(block: T.() -> Unit): T {\n

```

```

contract { \n    callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n block()\n return
this\n}\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns `this` value.\n *\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#also).\n
*\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic
inline fun <T> T.also(block: (T) -> Unit): T { \n    contract { \n        callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n    } \n    block(this)\n    return this\n}\n\n/**\n * Calls the specified function
[block] with `this` value as its argument and returns its result.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#let).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.let(block: (T) -> R): R { \n    contract { \n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    } \n    return block(this)\n}\n\n/**\n * Returns `this`
value if it satisfies the given [predicate] or `null`, if it doesn't.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-
takeunless).\n *\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic inline fun
<T> T.takeIf(predicate: (T) -> Boolean): T? { \n    contract { \n        callsInPlace(predicate,
InvocationKind.EXACTLY_ONCE)\n    } \n    return if (predicate(this)) this else null\n}\n\n/**\n * Returns `this`
value if it _does not_ satisfy the given [predicate] or `null`, if it does.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-
takeunless).\n *\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic inline fun <T>
T.takeUnless(predicate: (T) -> Boolean): T? { \n    contract { \n        callsInPlace(predicate,
InvocationKind.EXACTLY_ONCE)\n    } \n    return if (!predicate(this)) this else null\n}\n\n/**\n * Executes the
given function [action] specified number of [times].\n *\n * A zero-based index of current iteration is passed as a
parameter to [action].\n *\n * @sample samples.misc.ControlFlow.repeat\n *\n@kotlin.internal.InlineOnly\npublic
inline fun repeat(times: Int, action:
(Int) -> Unit) { \n    contract { callsInPlace(action) } \n\n    for (index in 0 until times) { \n        action(index)\n
}\n}\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@npackage kotlin.comparisons\n\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport
kotlin.js.*\n\n/**\n * Returns the greater of two values.\n *\n * If values are equal, returns the first one.\n
*\n@SinceKotlin("1.1")\npublic actual fun <T : Comparable<T>> maxOf(a: T, b: T): T { \n    return if (a >= b) a
else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Byte, b: Byte): Byte { \n
return maxOf(a.toInt(), b.toInt()).unsafeCast<Byte>()\n}\n\n/**\n
*\n * Returns the greater of two values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline
fun maxOf(a: Short, b: Short): Short { \n    return maxOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n
*\n * Returns the greater of two values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline
fun maxOf(a: Int, b: Int): Int { \n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.1")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun maxOf(a: Long, b:
Long): Long { \n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n *\n * If either value
is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
maxOf(a: Float, b: Float): Float { \n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*\n * If either value is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
actual inline fun maxOf(a: Double, b: Double): Double { \n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the
greater of three values.\n *\n * If there are multiple equal maximal values, returns the first of them.\n
*\n@SinceKotlin("1.1")\npublic actual fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T { \n    return
maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Byte, b: Byte, c: Byte):

```

```

Byte {
    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()
}

Returns the greater of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Short, b: Short, c: Short): Short {
    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()
}

Returns the greater of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Int, b: Int, c: Int): Int {
    return JsMath.max(a, b, c)
}

Returns the greater of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Long, b: Long, c: Long): Long {
    return maxOf(a, maxOf(b, c))
}

Returns the greater of three values.

* If any value is `NaN`, returns `NaN`.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Float, b: Float, c: Float): Float {
    return JsMath.max(a, b, c)
}

Returns the greater of three values.

* If any value is `NaN`, returns `NaN`.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Double, b: Double, c: Double): Double {
    return JsMath.max(a, b, c)
}

Returns the greater of the given values.

* If there are multiple equal maximal values, returns the first of them.

@SinceKotlin("1.4")
public actual fun <T : Comparable<T>>
    maxOf(a: T, vararg other: T): T {
        var max = a
        for (e in other) max = maxOf(max, e)
        return max
    }

Returns the greater of the given values.

@SinceKotlin("1.4")
public actual fun maxOf(a: Byte, vararg other: Byte): Byte {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the greater of the given values.

@SinceKotlin("1.4")
public actual fun maxOf(a: Short, vararg other: Short): Short {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the greater of the given values.

@SinceKotlin("1.4")
public actual fun maxOf(a: Int, vararg other: Int): Int {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the greater of the given values.

@SinceKotlin("1.4")
public actual fun maxOf(a: Long, vararg other: Long): Long {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the greater of the given values.

* If any value is `NaN`, returns `NaN`.

@SinceKotlin("1.4")
public actual fun maxOf(a: Float, vararg other: Float): Float {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the greater of the given values.

* If any value is `NaN`, returns `NaN`.

@SinceKotlin("1.4")
public actual fun maxOf(a: Double, vararg other: Double): Double {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

Returns the smaller of two values.

* If values are equal, returns the first one.

@SinceKotlin("1.1")
public actual fun <T : Comparable<T>> minOf(a: T, b: T): T {
    return if (a <= b) a else b
}

Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Byte, b: Byte): Byte {
    return minOf(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Short, b: Short): Short {
    return minOf(a.toInt(), b.toInt()).unsafeCast<Short>()
}

Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Int, b: Int): Int {
    return JsMath.min(a, b)
}

Returns the smaller of two values.

@SinceKotlin("1.1")
@Suppress("NOTHING_TO_INLINE")
public actual inline fun minOf(a: Long, b: Long): Long {
    return if (a <= b) a else b
}

Returns the smaller of two values.

* If either value is `NaN`, returns `NaN`.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Float, b: Float): Float {
    return JsMath.min(a, b)
}

Returns the smaller of two values.

* If either value is `NaN`, returns `NaN`.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Double, b: Double): Double {
    return JsMath.min(a, b)
}

Returns the smaller of three values.

* If there are multiple equal minimal values, returns the first of them.

@SinceKotlin("1.1")
public actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {
    return minOf(a, minOf(b, c))
}

Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {
    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()
}

Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun minOf(a: Short, b:

```

```

Short, c: Short): Short {\n    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()\n}\n\n/**\n *
Returns the smaller of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline
fun minOf(a: Int, b: Int, c: Int): Int {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three
values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Long, b: Long,
c: Long): Long {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
minOf(a: Float, b: Float, c: Float): Float {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three
values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double, c:
Double): Double {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If
there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin("1.4")\n\npublic actual fun <T
: Comparable<T>>
    minOf(a: T, vararg other: T): T {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return
min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a:
Byte, vararg other: Byte): Byte {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a: Short, vararg
other: Short): Short {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns
the smaller of the given values.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a: Int, vararg other: Int): Int
{\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the
given values.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a: Long, vararg other: Long): Long {\n    var
min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given
values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a: Float, vararg other: Float): Float {\n    var min = a\n    for (e
in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.4")\n\npublic actual fun minOf(a: Double, vararg other:
Double): Double {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n"/*\n * Copyright
2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\n\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\n\npublic value class ULong @PublishedApi
    internal constructor(@PublishedApi internal val data: Long) : Comparable<ULong> {\n\n    companion object {\n
        /**\n         * A constant holding the minimum value an instance of ULong can have.\n         */\n        public const
val MIN_VALUE: ULong = ULong(0)\n\n        /**\n         * A constant holding the maximum value an instance of
ULong can have.\n         */\n        public const val MAX_VALUE: ULong = ULong(-1)\n\n        /**\n         * The
number of bytes used to represent an instance of ULong in a binary form.\n         */\n        public const val
SIZE_BYTES: Int = 8\n\n        /**\n         * The number of bits used to represent an instance of ULong in a binary
form.\n         */\n        public const val SIZE_BITS: Int = 64\n    }\n\n    /**\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater
than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UByte): Int =
this.compareTo(other.toULong())\n\n    /**\n     * Compares this value with the specified value for order.\n     *
Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a
positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun
compareTo(other: UShort): Int = this.compareTo(other.toULong())\n\n    /**\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())\n\n    /**\n     *

```

Compares this value with the specified value for order.\n

```
* Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a\n * positive number if it's greater than other.\n * @kotlin.internal.InlineOnly\n @Suppress(\"OVERRIDE_BY_INLINE\")\n public override inline operator fun compareTo(other: ULong): Int =\n     ulongCompare(this.data, other.data)\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other:\n UShort): ULong = this.plus(other.toULong())\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other:\n ULong): ULong\n = ULong(this.data.plus(other.data))\n /** Subtracts the other value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): ULong =\n this.minus(other.toULong())\n /** Subtracts the other value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())\n /** Subtracts the other\n value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): ULong =\n this.minus(other.toULong())\n /** Subtracts the other value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))\n /** Multiplies\n this value by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte):\n ULong = this.times(other.toULong())\n /** Multiplies this value by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): ULong = this.times(other.toULong())\n /** Multiplies this\n value by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): ULong\n = this.times(other.toULong())\n /** Multiplies this value by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))\n /** Divides this\n value by the other value, truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UByte): ULong = this.div(other.toULong())\n /** Divides this value by the\n other value, truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public\n inline operator fun div(other: UShort): ULong = this.div(other.toULong())\n /** Divides this value by the other\n value, truncating the result\n to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UInt):\n ULong = this.div(other.toULong())\n /** Divides this value by the other value, truncating the result to an integer\n that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong =\n ulongDivide(this, other)\n /**\n * Calculates the remainder of truncating division of this value by the other\n value.\n * \n * The result is always less than the divisor.\n * @kotlin.internal.InlineOnly\n public\n inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())\n /**\n * Calculates the remainder\n of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n * @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): ULong =\n this.rem(other.toULong())\n /**\n * Calculates the remainder\n of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n * @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): ULong =\n this.rem(other.toULong())\n /**\n * Calculates the remainder of truncating division of this value by the other\n value.\n * \n * The result is always less than the divisor.\n * @kotlin.internal.InlineOnly\n public\n inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)\n /**\n * Divides this value by\n the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types,\n the results of flooring division and truncating division are the same.\n * @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value\n by the other value, flooring the result to an integer that is
```



```

represents the same numerical value as this `ULong`.
 * The resulting `Short` value is represented
by the least significant 16 bits of this `ULong` value.
 * Note that the resulting `Short` value may be negative.
 *
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort()
/**
 * Converts this
[ULong] value to [Int].
 *
 * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int`
value represents the same numerical value as this `ULong`.
 * The resulting `Int` value is
represented by the least significant 32 bits of this `ULong`
value.
 * Note that the resulting `Int` value may be negative.
 *
@kotlin.internal.InlineOnly
public
inline fun toInt(): Int = data.toInt()
/**
 * Converts this [ULong] value to [Long].
 *
 * If this value is
less than or equals to [Long.MAX_VALUE], the resulting `Long` value represents the same numerical value
as this `ULong`. Otherwise the result is negative.
 *
 * The resulting `Long` value has the same binary
representation as this `ULong` value.
 *
@kotlin.internal.InlineOnly
public inline fun toLong(): Long =
data
/**
 * Converts this [ULong] value to [UByte].
 *
 * If this value is less than or equals to
[UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `ULong`.
 *
 * The resulting `UByte` value is represented by the least significant 8 bits of this `ULong` value.
 *
@kotlin.internal.InlineOnly
public inline
fun toUByte(): UByte = data.toUByte()
/**
 * Converts this [ULong] value to [UShort].
 *
 * If this
value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same
numerical value as this `ULong`.
 *
 * The resulting `UShort` value is represented by the least significant 16
bits of this `ULong` value.
 *
@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort =
data.toUShort()
/**
 * Converts this [ULong] value to [UInt].
 *
 * If this value is less than or equals
to [UInt.MAX_VALUE], the resulting `UInt` value represents the same numerical value as this `ULong`.
 *
 * The resulting `UInt` value is represented by the least significant 32 bits of this `ULong` value.
 *
@kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = data.toUInt()
/** Returns this value.
 *
@kotlin.internal.InlineOnly
public inline fun toULong():
ULong = this
/**
 * Converts this [ULong] value to [Float].
 *
 * The resulting value is the closest
`Float` to this `ULong` value.
 * In case when this `ULong` value is exactly between two `Float`s,
 * the one
with zero at least significant bit of mantissa is selected.
 *
@kotlin.internal.InlineOnly
public inline fun
toFloat(): Float = this.toDouble().toFloat()
/**
 * Converts this [ULong] value to [Double].
 *
 * The
resulting value is the closest `Double` to this `ULong` value.
 * In case when this `ULong` value is exactly
between two `Double`s,
 * the one with zero at least significant bit of mantissa is selected.
 *
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)
public override
fun toString(): String = ulongToString(data)
/**
 * Converts this [Byte] value to [ULong].
 *
 * If this
value is positive, the resulting `ULong` value
represents the same numerical value as this `Byte`.
 * The least significant 8 bits of the resulting `ULong`
value are the same as the bits of this `Byte` value,
 * whereas the most significant 56 bits are filled with the sign bit
of this value.
 *
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun Byte.toULong(): ULong = ULong(this.toLong())
/**
 * Converts this [Short] value to
[ULong].
 *
 * If this value is positive, the resulting `ULong` value represents the same numerical value as this
`Short`.
 *
 * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short`
value,
 * whereas the most significant 48 bits are filled with the sign bit of this value.
 *
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun Short.toULong(): ULong = ULong(this.toLong())
/**
 * Converts this [Int]
value to [ULong].
 *
 * If this value is positive, the resulting `ULong` value represents the same numerical value
as this `Int`.
 *
 * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int`
value,
 * whereas the most significant 32 bits are filled with the sign bit of this value.
 *
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun Int.toULong(): ULong = ULong(this.toLong())
/**
 * Converts this [Long] value to [ULong].

```

```

*\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`. \n *\n
* The resulting `ULong` value has the same binary representation as this `Long` value.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun Long.toULong(): ULong = ULong(this)\n\n/**\n * Converts this [Float] value to [ULong].\n *\n * The fractional part,
if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`,
[ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())\n\n/**\n * Converts this [Double]
value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this
`Double` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun Double.toULong(): ULong = doubleToULong(this)\n", "/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1.\n
*\n@kotlin.internal.InlineOnly\nnpublic inline operator fun <T> List<T>.component1(): T {\n    return
get(0)\n}\n\n/**\n * Returns 2nd *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the
size of this list is less than 2.\n *\n@kotlin.internal.InlineOnly\nnpublic inline operator fun <T>
List<T>.component2(): T {\n    return get(1)\n}\n\n/**\n * Returns 3rd *element* from the list.\n *\n * Throws an
[IndexOutOfBoundsException] if the size of this list is less than 3.\n *\n@kotlin.internal.InlineOnly\nnpublic
inline operator fun <T> List<T>.component3(): T {\n    return get(2)\n}\n\n/**\n * Returns 4th *element* from the
list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.\n
*\n@kotlin.internal.InlineOnly\nnpublic inline operator fun <T> List<T>.component4(): T {\n    return
get(3)\n}\n\n/**\n * Returns 5th *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the
size of this list is less than 5.\n *\n@kotlin.internal.InlineOnly\nnpublic inline operator fun <T>
List<T>.component5(): T {\n    return get(4)\n}\n\n/**\n * Returns `true` if [element] is found in the collection.\n
*\n@npublic operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {\n    if
(this is Collection)\n        return contains(element)\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns an
element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this
collection.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n@npublic fun <T> Iterable<T>.elementAt(index: Int): T {\n
if (this is List)\n    return get(index)\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException("Collection doesn't contain element at index $index.") }\n}\n\n/**\n * Returns an
element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this list.\n
*\n * @sample samples.collections.Collections.Elements.elementAt\n *\n@kotlin.internal.InlineOnly\nnpublic
inline fun <T> List<T>.elementAt(index: Int): T {\n    return get(index)\n}\n\n/**\n * Returns an element at the
given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this collection.\n
*\n * @sample samples.collections.Collections.Elements.elementAtOrElse\n *\n@npublic fun <T>
Iterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (this is List)\n        return
this.getOrElse(index,
defaultValue)\n    if (index < 0)\n        return defaultValue(index)\n    val iterator = iterator()\n    var count = 0\n
while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return
element\n    }\n    return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of

```



```

calling the [defaultValue] function if the [index] is out of bounds of this list.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *^\n@kotlin.internal.InlineOnly\npublic inline fun
<T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this collection.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *^\npublic fun <T>
Iterable<T>.elementAtOrNull(index: Int): T? {\n    if (this
is List)\n        return this.getOrNull(index)\n    if (index < 0)\n        return null\n    val iterator = iterator()\n    var
count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    }\n    return null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is
out of bounds of this list.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n *^\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAtOrNull(index: Int): T? {\n    return
this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n *^\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {\n    return
firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate],
or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n *^\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.findLast(predicate: (T) -> Boolean): T? {\n    return
lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n *^\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.findLast(predicate: (T) -> Boolean): T? {\n    return
lastOrNull(predicate)\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the collection
is empty.\n * ^\npublic fun <T> Iterable<T>.first(): T {\n    when (this) {\n        is List -> return this.first()\n        else
-> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                throw
NoSuchElementException("Collection is empty.")\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the list is empty.\n * ^\npublic fun <T>
List<T>.first(): T {\n    if (isEmpty())\n        throw NoSuchElementException("List is empty.")\n    return
this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n * ^\npublic inline fun <T> Iterable<T>.first(predicate: (T)
-> Boolean): T {\n    for (element in this) if (predicate(element)) return element\n    throw
NoSuchElementException("Collection contains no element matching the predicate.")\n}\n\n/**\n * Returns the
first non-null value produced by [transform] function being applied to elements of this collection in iteration order,\n *
or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n *^\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOf(transform:
(T) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException("No element of the
collection was transformed to a non-null value.")\n}\n\n/**\n * Returns the first non-null value produced by
[transform] function being applied to elements of this collection in iteration order,\n * or `null` if no non-null value
was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n *^\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n * Returns the first element, or `null` if the collection is empty.\n * ^\npublic fun <T> Iterable<T>.firstOrNull(): T? {\n    when (this) {\n        is List -> {\n            if (isEmpty())\n                return null\n            else\n                return this[0]\n        }\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * Returns the
first element, or `null` if the list is empty.\n * ^\npublic fun <T> List<T>.firstOrNull(): T? {\n    return if (isEmpty())

```

```

null else this[0]\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not
found.\n * \npublic inline fun <T> Iterable<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n  for (element in this)
if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.getOrNull(index: Int, defaultValue: (Int) -> T): T
{\n  return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if
the [index] is out of bounds of this list.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n
*\npublic fun <T> List<T>.getOrNull(index: Int): T? {\n  return if (index >= 0 && index <= lastIndex) get(index)
else null\n}\n\n/**\n * Returns first index of [element], or -1 if the collection does not contain element.\n
*\npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.indexOf(element: T): Int {\n  if (this is List) return
this.indexOf(element)\n  var index = 0\n  for (item in this) {\n    checkIndexOverflow(index)\n    if (element
== item)\n      return index\n    index++\n  }\n  return -1\n}\n\n/**\n * Returns first index of [element], or -1
if the list does not contain element.\n
*\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false
warning, extension takes precedence in some cases\npublic fun <@kotlin.internal.OnlyInputTypes
T> List<T>.indexOf(element: T): Int {\n  return indexOf(element)\n}\n\n/**\n * Returns index of the first element
matching the given [predicate], or -1 if the collection does not contain such element.\n
*\npublic inline fun <T>
Iterable<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n  var index = 0\n  for (item in this) {\n
checkIndexOverflow(index)\n    if (predicate(item))\n      return index\n    index++\n  }\n  return -
1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the list does not contain
such element.\n
*\npublic inline fun <T> List<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n  var index = 0\n
for (item in this) {\n  if (predicate(item))\n    return index\n    index++\n  }\n  return -1\n}\n\n/**\n *
Returns index of the last element matching the given [predicate], or -1 if the collection does not contain such
element.\n
*\npublic inline fun
<T> Iterable<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n  var lastIndex = -1\n  var index = 0\n  for (item
in this) {\n    checkIndexOverflow(index)\n    if (predicate(item))\n      lastIndex = index\n    index++\n
}\n  return lastIndex\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the list
does not contain such element.\n
*\npublic inline fun <T> List<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n
val iterator = this.listIterator(size)\n  while (iterator.hasPrevious()) {\n    if (predicate(iterator.previous())) {\n
return iterator.nextIndex()\n    }\n  }\n  return -1\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the collection is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\npublic fun <T> Iterable<T>.last(): T {\n  when (this) {\n    is
List -> return this.last()\n    else -> {\n      val iterator
= iterator()\n      if (!iterator.hasNext())\n        throw NoSuchElementException("Collection is empty.")\n
      var last = iterator.next()\n      while (iterator.hasNext())\n        last = iterator.next()\n      return
last\n    }\n  }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the list is
empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\npublic fun <T> List<T>.last(): T {\n  if (isEmpty())\n    throw NoSuchElementException("List is empty.")\n  return this[lastIndex]\n}\n\n/**\n *
Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such
element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\npublic inline fun <T>
Iterable<T>.last(predicate: (T) -> Boolean): T {\n  var last: T? = null\n  var found = false\n  for (element in this)
{\n    if (predicate(element)) {\n      last = element\n      found = true\n    }\n  }\n  if (!found) throw
NoSuchElementException("Collection contains no element matching the predicate.")\n  @Suppress("UNCHECKED_CAST")\n  return last as T\n}\n\n/**\n * Returns the last element matching the
given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\npublic inline fun <T> List<T>.last(predicate: (T) -> Boolean): T
{\n  val iterator = this.listIterator(size)\n  while (iterator.hasPrevious()) {\n    val element = iterator.previous()\n

```

```

        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("List contains no element
matching the predicate.")\n}\n\n/**\n * Returns last index of [element], or -1 if the collection does not contain
element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.lastIndexOf(element: T): Int {\n    if
(this is List) return this.lastIndexOf(element)\n
    var lastIndex = -1\n    var index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (element
== item)\n            lastIndex = index\n            index++\n    }\n    return lastIndex\n}\n\n/**\n * Returns last index of
[element], or -1 if the list does not contain element.\n */\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases\npublic fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {\n    return
lastIndexOf(element)\n}\n\n/**\n * Returns the last element, or `null` if the collection is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Iterable<T>.lastOrNull(): T? {\n    when (this)
{\n        is List -> return if (isEmpty()) null else this[size - 1]\n        else -> {\n            val iterator = iterator()\n
if (!iterator.hasNext())\n                return null\n            var last = iterator.next()\n
            while (iterator.hasNext())\n                last = iterator.next()\n            return last\n        }\n    }\n}\n\n/**\n *
Returns the last element, or `null` if the list is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> List<T>.lastOrNull(): T? {\n    return if
(isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if
no such element was found.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun
<T> Iterable<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n    var last: T? = null\n    for (element in this) {\n
if (predicate(element)) {\n        last = element\n    }\n    }\n    return last\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T> List<T>.lastOrNull(predicate: (T)
-> Boolean): T? {\n    val iterator = this.listIterator(size)\n    while (iterator.hasPrevious()) {\n        val element =
iterator.previous()\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns a random
element from this collection.\n */\n * @throws NoSuchElementException if this collection is empty.\n */\n * @SinceKotlin("1.3")\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.random(): T {\n
return random(Random)\n}\n\n/**\n * Returns a random element from this collection using the specified source of
randomness.\n */\n * @throws NoSuchElementException if this collection is empty.\n */\n * @SinceKotlin("1.3")\n */\npublic fun <T> Collection<T>.random(random: Random): T {\n    if (isEmpty())\n        throw NoSuchElementException("Collection is empty.")\n    return elementAt(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this collection, or `null` if this collection is empty.\n */\n * @SinceKotlin("1.4")\n */\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Collection<T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this collection using the specified source of randomness, or `null` if this collection is empty.\n */\n * @SinceKotlin("1.4")\n */\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
Collection<T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return
elementAt(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the collection is
empty or has more than one element.\n */\n */\npublic fun <T> Iterable<T>.single(): T {\n    when (this) {\n        is List -
> return this.single()\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException("Collection is empty.")\n            val single = iterator.next()\n            if
(iterator.hasNext())\n                throw IllegalArgumentException("Collection has more than one element.")\n            return single\n        }\n    }\n}\n\n/**\n * Returns the single element, or throws an exception if the list is empty or has more than one
element.\n */\n */\npublic fun <T> List<T>.single(): T {\n    return when (size) {\n        0 -> throw
NoSuchElementException("List is empty.")\n        1 -> this[0]\n        else -> throw
IllegalArgumentException("List has more than one element.")\n    }\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\n */\npublic

```

```

inline fun <T> Iterable<T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Collection contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Collection contains no element matching the
    predicate.")
    @SuppressWarnings("UNCHECKED_CAST") return single as T
}

Returns single element, or `null` if the collection is empty or has more than one element.
public fun <T> Iterable<T>.singleOrNull(): T? {
    when (this) {
        is List -> return if (size == 1) this[0] else null
        else -> {
            val iterator = iterator()
            if (!iterator.hasNext()) return null
            val single = iterator.next()
            if (iterator.hasNext()) return null
            return single
        }
    }
}

Returns single element, or `null` if the list is empty or has more than one element.
public fun <T> List<T>.singleOrNull(): T? {
    return if (size == 1) this[0] else null
}

Returns the single element matching the given [predicate], or `null` if element was
not found or more than one element was found.
public inline fun <T> Iterable<T>.singleOrNull(predicate: (T) -> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

Returns a list containing all elements except first [n] elements.
@throws IllegalArgumentException if [n] is negative.
@sample
samples.collections.Collections.Transformations.drop

public fun <T> Iterable<T>.drop(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return toList()
    val list: ArrayList<T>
    if (this is Collection<*>) {
        val resultSize = size - n
        if (resultSize <= 0) return emptyList()
        if (resultSize == 1) return listOf(last())
        list = ArrayList<T>(resultSize)
        if (this is List<T>) {
            if (this is RandomAccess) {
                for (index in n until size) list.add(this[index])
            } else {
                for (item in listIterator()) list.add(item)
            }
            return list
        } else {
            list = ArrayList<T>()
            var count = 0
            for (item in this) {
                if (count >= n) list.add(item) else ++count
            }
            return list.optimizeReadOnlyList()
        }
    }
}

Returns a list containing all elements except last [n] elements.
@throws IllegalArgumentException if [n] is negative.
@sample
samples.collections.Collections.Transformations.drop

public fun <T> List<T>.dropLast(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

Returns a list containing all elements except last elements that satisfy the given [predicate].
@sample
samples.collections.Collections.Transformations.drop

public inline fun <T> List<T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {
    if (!isEmpty()) {
        val iterator = listIterator(size)
        while (iterator.hasPrevious()) {
            if (!predicate(iterator.previous())) return take(iterator.nextIndex() + 1)
        }
    }
    return emptyList()
}

Returns a list containing all elements except first elements that satisfy the given [predicate].
@sample
samples.collections.Collections.Transformations.drop

public inline fun <T> Iterable<T>.dropWhile(predicate: (T) -> Boolean): List<T> {
    var yielding = false
    val list = ArrayList<T>()
    for (item in this) {
        if (yielding) list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

Returns a list containing only elements matching the given [predicate].
@sample
samples.collections.Collections.Filtering.filter

public inline fun <T> Iterable<T>.filter(predicate: (T) -> Boolean): List<T> {
    return filterTo(ArrayList<T>(), predicate)
}

Returns a list containing only elements matching the given [predicate].
@param [predicate] function that takes the index of an element and the element itself
and returns the result of predicate evaluation on the element.
@sample
samples.collections.Collections.Filtering.filterIndexed

public inline fun <T> Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {
    return

```

```

filterIndexedTo(ArrayList<T>(), predicate)\n\n/**\n * Appends all elements matching the given [predicate] to
the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic
inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -
> Boolean): C {\n    forEachIndexed { index, element ->\n        if (predicate(index, element))
destination.add(element)\n    }\n    return destination\n}\n\n/**\n * Returns a list containing all elements that are
instances of specified type parameter R.\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R>
Iterable<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n    return
filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C :
MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination:
C): C {\n    for (element in this) if (element is R) destination.add(element)\n    return destination\n}\n\n/**\n *
Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Iterable<T>.filterNot(predicate: (T) ->
Boolean): List<T> {\n    return filterNotNullTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing all
elements that are not `null`.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n */\npublic
fun <T : Any> Iterable<T?>.filterNotNull(): List<T> {\n    return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n *
Appends all elements that are not `null` to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C : MutableCollection<in T>, T : Any>
Iterable<T?>.filterNotNullTo(destination: C): C {\n    for (element in this) if (element != null)
destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given
[destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C :
MutableCollection<in T>> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element
in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>>
Iterable<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if (predicate(element))
destination.add(element)\n    return destination\n}\n\n/**\n * Returns a list containing elements at indices in the
specified [indices] range.\n * \n * @sample\n */\npublic fun <T> List<T>.slice(indices: IntRange): List<T> {\n    if
(indices.isEmpty()) return
listOf()\n    return this.subList(indices.start, indices.endInclusive + 1).toList()\n}\n\n/**\n * Returns a list
containing elements at specified [indices].\n * \n * @sample\n */\npublic fun <T> List<T>.slice(indices: Iterable<Int>): List<T> {\n
val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =
ArrayList<T>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Iterable<T>.take(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (this
is Collection<T>) {\n        if (n >= size) return toList()\n        if (n == 1) return listOf(first())\n    }\n    var count = 0\n
val list = ArrayList<T>(n)\n    for (item in this)\n        {\n            list.add(item)\n            if (++count == n)\n                break\n        }\n    return
list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> List<T>.takeLast(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size =

```


equal elements preserve their order relative to each other after sorting.

```

public fun <T : Comparable<T>>
Iterable<T>.sortedDescending(): List<T> {
    return sortedWith(reverseOrder())
}

```

* Returns a list of all elements sorted according to the specified [comparator].

* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

```

public fun
<T> Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> {
    if (this is Collection) {
        if (size <= 1) return this.toList()
        @Suppress("UNCHECKED_CAST")
        return (toTypedArray<Any?>() as
Array<T>).apply { sortWith(comparator) }.asList()
    }
    return toMutableList().apply { sortWith(comparator) }
}

```

* Returns an array of Boolean containing all of the elements of this collection.

```

public fun
Collection<Boolean>.toBooleanArray(): BooleanArray {
    val result = BooleanArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Byte containing all of the elements of this collection.

```

public fun Collection<Byte>.toByteArray(): ByteArray {
    val result = ByteArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Char containing all of the elements of this collection.

```

public fun Collection<Char>.toCharArray(): CharArray {
    val result = CharArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Double containing all of the elements of this collection.

```

public fun Collection<Double>.toDoubleArray():
DoubleArray {
    val result = DoubleArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Float containing all of the elements of this collection.

```

public fun Collection<Float>.toFloatArray(): FloatArray {
    val result = FloatArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Int containing all of the elements of this collection.

```

public fun Collection<Int>.toIntArray(): IntArray {
    val result = IntArray(size)
    var index = 0
    for
(element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Long containing all of the elements of this collection.

```

public fun Collection<Long>.toLongArray(): LongArray {
    val result = LongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

```

* Returns an array of Short containing all of the elements of this collection.

```

public fun
Collection<Short>.toShortArray(): ShortArray {
    val result = ShortArray(size)
    var index = 0
    for (element
in this)
        result[index++] = element
    return result
}

```

* Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given collection.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original collection.

```

@sample samples.collections.Collections.Transformations.associate

```

```

public inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return
associateTo(LinkedHashMap<K, V>(capacity), transform)
}

```

* Returns a [Map] containing the elements from the given collection indexed by the key

* returned from [keySelector] function applied to each element.

* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

* The returned map preserves the entry iteration order of the original collection.

```

@sample
samples.collections.Collections.Transformations.associateBy

```

```

public inline fun <T, K>
Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity =
mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K,
T>(capacity), keySelector)
}

```

* Returns a [Map]

containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given collection.

* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

* The returned map preserves the entry iteration order of the original collection.

```

@sample samples.collections.Collections.Transformations.associateByWithValueTransform

```

```

public inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return
associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

```

* Populates and

returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given collection and value is the element itself. If any two elements would have the same

key returned by [keySelector] the last one gets added to the map. @sample

```
samples.collections.Collections.Transformations.associateByTo\n\npublic inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n    for (element in this) {\n        destination.put(keySelector(element), element)\n    }\n    return destination\n}\n\nPopulates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given collection. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample
```

```
samples.collections.Collections.Transformations.associateByToWithValueTransform\n\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n    for (element in this) {\n        destination.put(keySelector(element), valueTransform(element))\n    }\n    return destination\n}\n\nPopulates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given collection. If any of two pairs would have the same key the last one gets added to the map. @sample
```

```
samples.collections.Collections.Transformations.associateTo\n\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n    for (element in this) {\n        destination += transform(element)\n    }\n    return destination\n}\n\nReturns a [Map] where keys are elements from the given collection and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves
```

the entry iteration order of the original collection. @sample

```
samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.3")\npublic inline fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n    val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nPopulates and returns the [destination] mutable map with key-value pairs for each element of the given collection, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map. @sample
```

```
samples.collections.Collections.Transformations.associateWithTo\n\n@SinceKotlin("1.3")\npublic inline fun <K, V, M : MutableMap<in K, in V>>
```

```
Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\nAppends all elements to the given [destination] collection. @sample
```

```
samples.collections.Collections.Transformations.toCollection\n\npublic fun <T, C : MutableCollection<in T>> Iterable<T>.toCollection(destination: C): C {\n    for (item in this) {\n        destination.add(item)\n    }\n    return destination\n}\n\nReturns a new [HashSet] of all elements. @sample
```

```
samples.collections.Collections.Transformations.toHashSet\n\npublic fun <T> Iterable<T>.toHashSet(): HashSet<T> {\n    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))\n}\n\nReturns a [List] containing all elements. @sample
```

```
samples.collections.Collections.Transformations.toList\n\npublic fun <T> Iterable<T>.toList(): List<T> {\n    if (this is Collection) {\n        return when (size) {\n            0 -> emptyList()\n            1 -> listOf(if (this is List) get(0) else iterator().next())\n            else -> this.toMutableList()\n        }\n    }\n    return this.toMutableList().optimizeReadOnlyList()\n}\n\nReturns a new [MutableList] filled with all elements of this collection. @sample
```

```
samples.collections.Collections.Transformations.toMutableList\n\npublic fun <T> Iterable<T>.toMutableList(): MutableList<T> {\n    if (this is Collection<T>) {\n        return this.toMutableList()\n    }\n    return toCollection(ArrayList<T>())\n}\n\nReturns a new [MutableList] filled with all elements of this collection. @sample
```

```
samples.collections.Collections.Transformations.toMutableList\n\npublic fun <T> Collection<T>.toMutableList(): MutableList<T> {\n    return ArrayList(this)\n}\n\nReturns a [Set] of all elements. The returned set preserves the element iteration order of the original collection. @sample
```

```
samples.collections.Collections.Transformations.toSet\n\npublic fun <T> Iterable<T>.toSet(): Set<T> {\n    if (this is
```



```

MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original collection by the key returned by the given [keySelector]
 * function applied to each element and returns a map where each group key is associated with a list of
 * corresponding elements.
 *
 * The returned map preserves the entry iteration order of the keys produced from the
 * original collection.
 *
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K,
    List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function applied to the element and returns a map where each group key is
 * associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the
 * keys produced from the original collection.
 *
 * @sample
 * samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <T, K, V>
    Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {
    return
        groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups elements
 * of the original collection by the key returned by the given [keySelector] function
 * applied to each element and
 * puts to the [destination] map each group key associated with a list
 * of corresponding elements.
 *
 * @return The [destination] map.
 *
 * @sample
 * samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, M : MutableMap<in K,
    MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function
 * applied to each element of the original collection
 * by the key returned by the given [keySelector] function applied
 * to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample
 * samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <T, K, V, M :
    MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination:
    M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key =
            keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Creates a [Grouping] source from a
 * collection to be used later with one of group-and-fold operations
 * using the specified [keySelector] function to
 * extract a key from each element.
 *
 * @sample samples.collections.Grouping.groupingByEachCount
 */
@SinceKotlin("1.1")
public inline fun <T, K> Iterable<T>.groupingBy(crossinline keySelector: (T) -> K):
    Grouping<T, K> {
    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> =
            this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original
 * collection.
 *
 * @sample samples.collections.Collections.Transformations.map
 */
public inline fun <T, R>
    Iterable<T>.map(transform: (T) -> R): List<R> {
    return mapTo(ArrayList<R>(collectionSizeOrDefault(10)),
        transform)
}

/**
 * Returns a list containing the results of applying the given [transform] function
 * to each
 * element and its index in the original collection.
 *
 * @param [transform] function that takes the index of an element
 * and the element itself
 * and returns the result of the transform applied to the element.
 */
public inline fun <T,
    R> Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {
    return
        mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

/**
 * Returns a list containing only
 * the non-null results of applying the given [transform] function
 * to each element and its index in the original
 * collection.
 *
 * @param [transform] function that takes the index of an element and the element itself
 * and returns
 * the result
 */

```

of the transform applied to the element.

```

public inline fun <T, R : Any>
Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {
    return
    mapIndexedNotNullTo(ArrayList<R>(), transform)
}

```

Applies the given [transform] function to each element and its index in the original collection and appends only the non-null results to the given [destination].

@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

```

public inline fun <T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) }
}
    return destination
}

```

Applies the given [transform] function to each element and its index in the original collection and appends the results to the given [destination].

@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

```

public inline fun <T, R, C : MutableCollection<in R>>
Iterable<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}

```

Returns a list containing only the non-null results of applying the given [transform] function to each element in the original collection.

@sample samples.collections.Collections.Transformations.mapNotNull

```

public inline fun <T, R : Any>
Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
    return
    mapNotNullTo(ArrayList<R>(), transform)
}

```

Applies the given [transform] function to each element in the original collection and appends only the non-null results to the given [destination].

```

public inline fun <T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) }
}
    return destination
}

```

Applies the given [transform] function to each element of the original collection and appends the results to the given [destination].

```

public inline fun <T, R, C : MutableCollection<in R>>
Iterable<T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

```

Returns a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

```

public fun <T>
Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

```

Returns a list containing only distinct elements from the given collection.

Among equal elements of the given collection, only the first one will be present in the resulting list.

The elements in the resulting list are in the same order as they were in the source collection.

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

public fun <T>
Iterable<T>.distinct(): List<T> {
    return this.toMutableSet().toList()
}

```

Returns a list containing only elements from the given collection having distinct keys returned by the given [selector] function.

Among elements of the given collection with equal keys, only the first one will be present in the resulting list.

The elements in the resulting list are in the same order as they were in the source collection.

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

public inline fun <T, K>
Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
    val set = HashSet<K>()
    val list = ArrayList<T>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

```

Returns a set containing all elements that are contained by both this collection and the specified collection.

The returned set preserves the element iteration order of the original collection.

To get a set containing all elements that are contained at least in one of these collections use [union].

```

public infix fun <T>
Iterable<T>.intersect(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

```

Returns a set containing all elements that are contained by this collection and not contained by the specified collection.

The returned set preserves the element iteration order of the original collection.

```

public infix fun <T>
Iterable<T>.subtract(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.removeAll(other)
}

```



```

element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun
<T, R> List<T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {\n    var accumulator =
initial\n    if (!isEmpty()) {\n        val iterator = listIterator(size)\n        while (iterator.hasPrevious()) {\n            val
index = iterator.previousIndex()\n            accumulator = operation(index, iterator.previous(), accumulator)\n        }\n    }\n    return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n */\n@kotlin.internal.HidesMembers\npublic inline fun <T> Iterable<T>.forEach(action: (T) -> Unit): Unit {\n    for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential
index with the element.\n * @param [action] function that takes the index of an element and the element itself\n *
and performs the action on the element.\n */\npublic inline fun <T> Iterable<T>.forEachIndexed(action:
(index: Int, T) -> Unit): Unit {\n    var index = 0\n    for (item in this) action(checkIndexOverflow(index++),
item)\n}\n\n@Deprecated("Use maxOrNull instead.")\nReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Double>.max(): Double? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.")\nReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Float>.max(): Float? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.")\nReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Iterable<T>.max(): T? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.")\nReplaceWith("this.maxByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince
= "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.maxBy(selector: (T) -> R): T? {\n    return maxByOrNull(selector)\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var maxElem = iterator.next()\n    if (!iterator.hasNext()) return maxElem\n    var
maxValue = selector(maxElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (maxValue <
v) {\n            maxElem = e\n            maxValue = v\n        }\n    } while (iterator.hasNext())\n    return
maxElem\n}\n\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result
is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws
NoSuchElementException if the collection is empty.\n */

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.maxOf(selector:
(T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if
(maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are
no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)
-> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var maxValue =
selector(iterator.next())\n    while (iterator.hasNext())
{\n        val v = selector(iterator.next())\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)
-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var maxValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        maxValue =
maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each
element in the collection or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.maxOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the collection.\n * \n * @throws
NoSuchElementException if the collection is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator
= iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var maxValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the collection or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Iterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic fun Iterable<Double>.maxOrNull(): Double? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e =

```

```

iterator.next()\n    max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if
there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun
Iterable<Float>.maxOrNull(): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max
= iterator.next()\n    while (iterator.hasNext())
{\n        val e = iterator.next()\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest
element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>>
Iterable<T>.maxOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max =
iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (max < e) max = e\n    }\n
return max\n}\n\n@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T? {\n
return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T>
Iterable<T>.maxWithOrNull(comparator:
Comparator<in T>): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max =
iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (comparator.compare(max, e) <
0) max = e\n    }\n    return max\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Double>.min(): Double? {\n    return
minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Float>.min(): Float? {\n    return
minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun
<T : Comparable<T>> Iterable<T>.min(): T? {\n    return minOrNull()\n}\n\n@Deprecated("Use minByOrNull
instead.", ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.minBy(selector: (T) -> R): T? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var minElem = iterator.next()\n    if (!iterator.hasNext()) return minElem\n    var
minValue = selector(minElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (minValue >
v) {\n            minElem
= e\n            minValue = v\n        }\n    } while (iterator.hasNext())\n    return minElem\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector]
function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is
`NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws
NoSuchElementException if the collection is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Comparable<R>> Iterable<T>.minOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (minValue > v) {\n           
minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)
-> Double): Double?
{\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n
while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n
return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)
-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n       
minValue =
minOf(minValue,
v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each element in the collection or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.minOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the collection.\n * \n * @throws
NoSuchElementException if the collection is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R> Iterable<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator
= iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the collection or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Iterable<T>.minOfWithOrNull(comparator:

```



```

Comparator<in R>, selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @SinceKotlin("1.4")\npublic fun Iterable<Double>.minOrNull(): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =\n        iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if\nthere are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @SinceKotlin("1.4")\npublic fun\nIterable<Float>.minOrNull():\nFloat? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while\n(iterator.hasNext()) {\n        val e = iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * @SinceKotlin("1.4")\npublic fun <T :\nComparable<T>> Iterable<T>.minOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (min > e) min = e\n    }\n    return min\n}\n\n@Deprecated("Use minWithOrNull instead.")\nReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\npublic fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T? {\n    return minWithOrNull(comparator)\n}\n\nReturns the first element having the smallest\nvalue according to the provided [comparator] or `null` if there are no elements.\n * \n * @SinceKotlin("1.4")\npublic\nfun <T> Iterable<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n    val iterator = iterator()\n    if\n(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =\n        iterator.next()\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\nReturns `true` if\nthe collection has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n * \n * @public fun\n<T> Iterable<T>.none(): Boolean {\n    if (this is Collection) return isEmpty()\n    return\n!iterator().hasNext()\n}\n\nReturns `true` if no elements match the given [predicate].\n * \n * @sample\nsamples.collections.Collections.Aggregates.noneWithPredicate\n * \n * @public inline fun <T>\nIterable<T>.none(predicate: (T) -> Boolean): Boolean {\n    if (this is Collection && isEmpty()) return true\n    for\n(element\nin this) if (predicate(element)) return false\n    return true\n}\n\nPerforms the given [action] on each element\nand returns the collection itself afterwards.\n * \n * @SinceKotlin("1.1")\npublic inline fun <T, C : Iterable<T>>\nC.onEach(action: (T) -> Unit): C {\n    return apply { for (element in this) action(element) }\n}\n\nPerforms\nthe given [action] on each element, providing sequential index with the element,\n * and returns the collection itself\nafterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs\nthe action on the element.\n * \n * @SinceKotlin("1.4")\npublic inline fun <T, C : Iterable<T>>\nC.onEachIndexed(action: (index: Int, T) -> Unit): C {\n    return apply { forEachIndexed(action) }\n}\n\nAccumulates value starting with the first element and applying [operation] from left to right\n * to current\naccumulator value and each element.\n * \n * Throws an exception if this collection is empty.\n\nIf the collection can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its\nreceiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * \n * @public inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {\n    val iterator = this.iterator()\n    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")\n    var\naccumulator: S = iterator.next()\n    while (iterator.hasNext()) {\n        accumulator = operation(accumulator,\n        iterator.next())\n    }\n    return accumulator\n}\n\nAccumulates value starting with the first element and\napplying [operation] from left to right\n * to current accumulator value and each element with its index in the\noriginal collection.\n * \n * Throws an exception if this collection is empty.\n\nIf the collection can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null`\nwhen its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current

```

accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduce
public inline fun <S, T : S>
Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator =
            operation(checkIndexOverflow(index++), accumulator, iterator.next())
    }
    return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original collection.

Returns `null` if the collection is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceOrNull
@SinceKotlin("1.4")
public inline fun <S, T : S> Iterable<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator =
            operation(checkIndexOverflow(index++), accumulator, iterator.next())
    }
    return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.

Returns `null` if the collection is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceOrNull
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S>
Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator =
            operation(accumulator, iterator.next())
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.

Throws an exception if this list is empty. If the list can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceRight
public inline fun <S, T : S>
List<T>.reduceRight(operation: (T, acc: S) -> S): S {
    val iterator = listIterator(size)
    if (!iterator.hasPrevious())
        throw UnsupportedOperationException("Empty list can't be reduced.")
    var accumulator: S = iterator.previous()
    while (iterator.hasPrevious()) {
        accumulator =
            operation(iterator.previous(), accumulator)
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original list and current accumulator value.

Throws an exception if this list is empty. If the list can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceRight
public inline fun <S, T : S>
List<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {
    val iterator = listIterator(size)
    if (!iterator.hasPrevious())
        throw UnsupportedOperationException("Empty list can't be reduced.")
    var accumulator: S = iterator.previous()
    while (iterator.hasPrevious()) {
        val index = iterator.previousIndex()
        accumulator = operation(index, iterator.previous(), accumulator)
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original list and current accumulator value.

Returns `null` if the list is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceRightOrNull

```

```

*\\n@SinceKotlin("\\1.4\\")\\npublic inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\\n    val iterator = listIterator(size)\\n    if (!iterator.hasPrevious())\\n        return null\\n    var accumulator: S = iterator.previous()\\n    while (iterator.hasPrevious()) {\\n        val index = iterator.previousIndex()\\n        accumulator = operation(index, iterator.previous(), accumulator)\\n    }\\n    return accumulator\\n}\\n}\\n/n/**\\n * Accumulates value starting with the last element and applying [operation] from right to left\\n * to each element and current accumulator value.\\n * \\n * Returns `null` if the list is empty.\\n * \\n * @param [operation] function that takes an element and current accumulator value,\\n * and calculates the next accumulator value.\\n * \\n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\\n
*\\n@SinceKotlin("\\1.4\\")\\n@WasExperimental(ExperimentalStdlibApi::class)\\npublic inline fun <S, T : S> List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\\n    val iterator = listIterator(size)\\n    if (!iterator.hasPrevious())\\n        return null\\n    var accumulator: S = iterator.previous()\\n    while (iterator.hasPrevious()) {\\n        accumulator = operation(iterator.previous(), accumulator)\\n    }\\n    return accumulator\\n}\\n}\\n/n/**\\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\\n * to each element and current accumulator value that starts with [initial] value.\\n * \\n * Note that `acc` value passed to [operation] function should not be mutated;\\n * otherwise it would affect the previous value in resulting list.\\n * \\n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\\n * \\n * @sample samples.collections.Collections.Aggregates.runningFold\\n
*\\n@SinceKotlin("\\1.4\\")\\npublic inline fun <T, R> Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\\n    val estimatedSize = collectionSizeOrDefault(9)\\n    if (estimatedSize == 0) return listOf(initial)\\n    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\\n    var accumulator = initial\\n    for (element in this) {\\n        accumulator = operation(accumulator, element)\\n        result.add(accumulator)\\n    }\\n    return result\\n}\\n}\\n/n/**\\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\\n * to each element, its index in the original collection and current accumulator value that starts with [initial] value.\\n * \\n * Note that `acc` value passed to [operation] function should not be mutated;\\n * otherwise it would affect the previous value in resulting list.\\n * \\n * @param [operation] function that takes the index of an element, current accumulator value\\n * and the element itself, and calculates the next accumulator value.\\n * \\n * @sample samples.collections.Collections.Aggregates.runningFold\\n
*\\n@SinceKotlin("\\1.4\\")\\npublic inline fun <T, R> Iterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\\n    val estimatedSize = collectionSizeOrDefault(9)\\n    if (estimatedSize == 0) return listOf(initial)\\n    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\\n    var index = 0\\n    var accumulator = initial\\n    for (element in this) {\\n        accumulator = operation(index++, accumulator, element)\\n        result.add(accumulator)\\n    }\\n    return result\\n}\\n}\\n/n/**\\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\\n * to each element and current accumulator value that starts with the first element of this collection.\\n * \\n * Note that `acc` value passed to [operation] function should not be mutated;\\n * otherwise it would affect the previous value in resulting list.\\n * \\n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\\n * \\n * @sample samples.collections.Collections.Aggregates.runningReduce\\n
*\\n@SinceKotlin("\\1.4\\")\\n@WasExperimental(ExperimentalStdlibApi::class)\\npublic inline fun <S, T : S> Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\\n    val iterator = this.iterator()\\n    if (!iterator.hasNext()) return emptyList()\\n    var accumulator: S = iterator.next()\\n    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\\n    while (iterator.hasNext()) {\\n        accumulator = operation(accumulator, iterator.next())\\n        result.add(accumulator)\\n    }\\n    return result\\n}\\n}\\n/n/**\\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\\n * to each element, its index in the original collection and current accumulator value that starts with the first element of this collection.\\n * \\n * Note that `acc` value passed to [operation] function should not be mutated;\\n * otherwise

```

it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <S, T : S> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return emptyList()\n var accumulator: S = iterator.next()\n val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\n var index = 1\n while (iterator.hasNext()) {\n accumulator = operation(index++, accumulator, iterator.next())\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun <T, R> Iterable<T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun <T, R> Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n * \n * @Deprecated("Use sumOf instead.")\n * \n * ReplaceWith("this.sumOf(selector)")\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public inline fun <T> Iterable<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n * \n * @Deprecated("Use sumOf instead.")\n * \n * ReplaceWith("this.sumOf(selector)")\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public inline fun <T> Iterable<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n * \n * @SinceKotlin("1.4")\n * \n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n * @OverloadResolutionByLambdaReturnType\n * \n * @kotlin.jvm.JvmName("sumOfDouble")\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n * \n * @SinceKotlin("1.4")\n * \n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n * @OverloadResolutionByLambdaReturnType\n * \n * @kotlin.jvm.JvmName("sumOfInt")\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n * \n * @SinceKotlin("1.4")\n * \n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n * @OverloadResolutionByLambdaReturnType\n * \n * @kotlin.jvm.JvmName("sumOfLong")\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Long): Long {\n

```

    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/** Returns the sum of all values produced by [selector] function applied to each element in the collection.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("sumOfUInt")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/** Returns the sum of all values produced by [selector] function applied to each element in the collection.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/** Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.

public fun <T : Any> Iterable<T?>.requireNotNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}

/** Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.

public fun <T : Any> List<T?>.requireNotNulls(): List<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as List<T>
}

/** Splits this collection into a list of lists each not exceeding the given [size].

* The last list in the resulting list may have fewer elements than the given [size].

* @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.

* @sample samples.collections.Collections.Transformations.chunked

@SinceKotlin("1.2")
public fun <T> Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}

/** Splits this collection into several lists each not exceeding the given [size] and applies the given [transform] function to an each.

* @return list of results of the [transform] applied to an each list.

* Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.

* You should not store it or allow it to escape in some way, unless you made a snapshot of it.

* The last list may have fewer elements than the given [size].

* @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.

* @sample samples.text.Strings.chunkedTransform

@SinceKotlin("1.2")
public fun <T, R> Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

/** Returns a list containing all elements of the original collection without the first occurrence of the given [element].

public operator fun <T> Iterable<T>.minus(element: T): List<T> {
    val result = ArrayList<T>(collectionSizeOrDefault(10))
    var removed = false
    return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }
}

/** Returns a list containing all elements of the original collection except the elements contained in the given [elements] array.

* Before Kotlin 1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have a correct and stable implementation of `hashCode()` that didn't change between successive invocations.

* On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.

public operator fun <T> Iterable<T>.minus(elements: Array<out T>): List<T> {
    if (elements.isEmpty()) return this.toList()
    val other = elements.convertToSetForSetOperation()
    return this.filterNot { it in other }
}

/** Returns a list containing all elements of the original collection except the elements contained in the given [elements] collection.

```

```

* \n * Before Kotlin 1.6, the [elements] collection may have been converted to a [HashSet] to
speed up the operation, thus the elements were required to have\n * a correct and stable implementation of
`hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back
with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\npublic operator
fun <T> Iterable<T>.minus(elements: Iterable<T>): List<T> {\n    val other =
elements.convertToSetForSetOperationWith(this)\n    if (other.isEmpty())\n        return this.toList()\n    return
this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the
elements contained in the given [elements] sequence.\n * \n * Before Kotlin 1.6, the [elements] sequence may have
been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and
stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can
enable this behavior
back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\npublic
operator fun <T> Iterable<T>.minus(elements: Sequence<T>): List<T> {\n    val other =
elements.convertToSetForSetOperation()\n    if (other.isEmpty())\n        return this.toList()\n    return this.filterNot {
it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence
of the given [element].\n * \n *@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.minusElement(element: T): List<T> {\n    return minus(element)\n}\n\n/**\n * Splits the original
collection into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while
*second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Iterables.Operations.partition\n *\npublic inline fun <T> Iterable<T>.partition(predicate: (T) ->
Boolean): Pair<List<T>, List<T>> {\n
    val first = ArrayList<T>()\n    val second = ArrayList<T>()\n    for (element in this) {\n        if
(predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n
return Pair(first, second)\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the
given [element].\n * \npublic operator fun <T> Iterable<T>.plus(element: T): List<T> {\n    if (this is Collection)
return this.plus(element)\n    val result = ArrayList<T>()\n    result.addAll(this)\n    result.add(element)\n    return
result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n
*\npublic operator fun <T> Collection<T>.plus(element: T): List<T> {\n    val result = ArrayList<T>(size + 1)\n
result.addAll(this)\n    result.add(element)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the
original collection and then all elements of the
given [elements] array.\n * \npublic operator fun <T> Iterable<T>.plus(elements: Array<out T>): List<T> {\n    if
(this is Collection) return this.plus(elements)\n    val result = ArrayList<T>()\n    result.addAll(this)\n
result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] array.\n * \npublic operator fun <T>
Collection<T>.plus(elements: Array<out T>): List<T> {\n    val result = ArrayList<T>(this.size + elements.size)\n
result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then all elements of the given [elements] collection.\n * \npublic operator fun <T>
Iterable<T>.plus(elements: Iterable<T>): List<T> {\n    if (this is Collection) return this.plus(elements)\n    val
result = ArrayList<T>()\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n
* Returns a list containing all elements of the original collection and then all elements of the given [elements]
collection.\n * \npublic operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {\n    if (elements is
Collection) {\n        val result = ArrayList<T>(this.size + elements.size)\n        result.addAll(this)\n
result.addAll(elements)\n        return result\n    } else {\n        val result = ArrayList<T>(this)\n
result.addAll(elements)\n        return result\n    }\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n * \npublic operator fun <T>
Iterable<T>.plus(elements: Sequence<T>): List<T> {\n    val result = ArrayList<T>()\n    result.addAll(this)\n
result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n * \npublic operator

```

```

fun <T> Collection<T>.plus(elements: Sequence<T>): List<T> {
    val result = ArrayList<T>(this.size + 10)
    result.addAll(this)
    result.addAll(elements)
    return result
}

// Returns a list containing all elements
// of the original collection and then the given [element].
// @kotlin.internal.InlineOnly
public inline fun <T>
Iterable<T>.plusElement(element: T): List<T> {
    return plus(element)
}

// Returns a list containing all
// elements of the original collection and then the given [element].
// @kotlin.internal.InlineOnly
public inline fun
<T> Collection<T>.plusElement(element: T): List<T> {
    return plus(element)
}

// Returns a list of
// snapshots of the window of the given [size]
// sliding along this collection with the given [step], where each
// snapshot is a list.
// Several last lists may have fewer elements than the given [size].
// Both [size] and
// [step] must be positive and can be greater than the number of elements
// in this collection.
// @param size the number of elements to take in each window
// @param step the number of
// elements to move the window forward by on an each step, by default 1
// @param partialWindows controls
// whether or not to keep partial windows in the end if any,
// by default `false` which means partial windows won't
// be preserved
// @sample samples.collections.Sequences.Transformations.takeWindows
// @SinceKotlin("1.2")
public fun <T> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean
= false): List<List<T>> {
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize = this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result = ArrayList<List<T>>(resultCapacity)
        var index = 0
        while (index in 0 until thisSize) {
            val
            windowSize = size.coerceAtMost(thisSize - index)
            if (windowSize < size && !partialWindows)
                break
            result.add(List(windowSize) { this[it + index] })
            index += step
        }
        return result
    }
    val result = ArrayList<List<T>>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer =
false).forEach {
        result.add(it)
    }
    return result
}

// Returns a list of results of applying the
// given [transform] function to
// an each list representing a view over the window of the given [size]
// sliding
// along this collection with the given [step].
// Note that the list passed to the [transform] function is ephemeral
// and is valid only inside that function.
// You should not store it or allow it to escape in some way, unless you made
// a snapshot of it.
// Several last lists may have fewer elements than the given [size].
// Both [size] and [step]
// must be positive and can be greater than the number of elements in this collection.
// @param size the number of
// elements to take in each
// window
// @param step the number of elements to move the window forward by on an each step, by default 1
// @param partialWindows controls whether or not to keep partial windows in the end if any,
// by default `false`
// which means partial windows won't be preserved
// @sample
samples.collections.Sequences.Transformations.averageWindows
// @SinceKotlin("1.2")
public fun <T, R>
Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R):
List<R> {
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize =
        this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result =
        ArrayList<R>(resultCapacity)
        val window = MovingSubList(this)
        var index = 0
        while (index in 0
        until thisSize) {
            val windowSize = size.coerceAtMost(thisSize - index)
            if (!partialWindows &&
            windowSize < size) break
            window.move(index, index + windowSize)
            result.add(transform(window))
            index += step
        }
        return result
    }
    val result = ArrayList<R>()
    windowedIterator(iterator(), size, step,
    partialWindows, reuseBuffer = true).forEach {
        result.add(transform(it))
    }
    return result
}

// Returns a list of pairs built from the elements of `this` collection and the [other] array with the same index.
// The
// returned list has length of the shortest collection.
// @sample
samples.collections.Iterables.Operations.zipIterable
// @public infix fun <T, R> Iterable<T>.zip(other: Array<out
R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

// Returns a list of values built from
// the elements of `this` collection and the [other] array with the same index
// using the provided [transform]
// function applied to each pair of elements.
// The returned list has length of the shortest collection.
// @sample
samples.collections.Iterables.Operations.zipIterableWithTransform
// @public inline fun <T, R,
V> Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = other.size
}

```

```

val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in this) {\n
if (i >= arraySize) break\n  list.add(transform(element, other[i++]))\n  }\n  return list\n}\n\n/**\n * Returns a
list of pairs built from the elements of `this` collection and [other] collection with the same index.\n * The returned
list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*/\npublic infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 ->
t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` collection and the [other] collection
with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V>
Iterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n  val first = iterator()\n  val second
= other.iterator()\n  val list = ArrayList<V>(minOf(collectionSizeOrDefault(10),
other.collectionSizeOrDefault(10)))\n  while (first.hasNext() && second.hasNext()) {\n
list.add(transform(first.next(), second.next()))\n  }\n  return list\n}\n\n/**\n * Returns a list of pairs of each two
adjacent elements in this collection.\n * \n * The returned list is empty if this collection contains less than two
elements.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n
*/\n@SinceKotlin("1.2")\npublic fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {\n  return zipWithNext
{ a, b -> a to b }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent
elements in this collection.\n * \n * The returned list is empty if this collection contains less than two elements.\n *
\n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n
*/\n@SinceKotlin("1.2")\npublic inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> R):
List<R> {\n  val iterator = iterator()\n  if (!iterator.hasNext()) return emptyList()\n  val result =
mutableListOf<R>()\n  var current = iterator.next()\n  while (iterator.hasNext()) {\n    val next =
iterator.next()\n    result.add(transform(current, next))\n    current = next\n  }\n  return result\n}\n\n/**\n *
Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if
supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit],
in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults
to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <T, A :
Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = "\", \", prefix: CharSequence = "\", postfix:
CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A
{\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      buffer.appendElement(element, transform)\n
    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n
return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n */\npublic fun <T>
Iterable<T>.joinToString(separator: CharSequence = "\", \", prefix: CharSequence = "\", postfix: CharSequence =
\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {\n  return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Returns this
collection as an [Iterable].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.asIterable():
Iterable<T> {\n  return this\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original collection returning
its elements when being iterated.\n * \n * @sample
samples.collections.Sequences.Building.sequenceFromCollection\n */\npublic fun
<T> Iterable<T>.asSequence(): Sequence<T> {\n  return Sequence { this.iterator() }\n}\n\n/**\n * Returns an
average value of elements in the collection.\n */\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun
Iterable<Byte>.average(): Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n

```



```

sum += element\n    checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*/\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Iterable<Short>.average(): Double {\n    var sum:
Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun
Iterable<Int>.average(): Double {\n    var sum: Double
= 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*/\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun
Iterable<Long>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n
sum += element\n    checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*/\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Iterable<Float>.average(): Double {\n    var sum:
Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName("averageOfDouble")\npublic
fun Iterable<Double>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this)
{\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum
/ count\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Iterable<Byte>.sum(): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the
collection.\n */\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Iterable<Short>.sum(): Int {\n    var sum: Int
= 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all
elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Iterable<Int>.sum(): Int {\n    var
sum: Int = 0\n    for (element in this) {\n        sum += element\n
    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Iterable<Long>.sum(): Long {\n    var sum: Long = 0L\n
for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in
the collection.\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Iterable<Float>.sum(): Float {\n    var
sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the
sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun
Iterable<Double>.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += element\n
    }\n    return sum\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.collections\n\nimport kotlin.comparisons.naturalOrder\nimport kotlin.random.Random\n\n/**\n * Returns
the array if it's not `null`, or an empty array otherwise.\n */\n@sample
samples.collections.Arrays.Usage.arrayOrEmpty\n\n@kotlin.internal.InlineOnly\npublic actual inline fun <T>
Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()\n\n/**\n * Returns a *typed* array containing
all of the elements of this collection.\n */\n\n/**\n * Allocates an array of runtime type `T` having its size equal to the size
of this collection\n */\n * and populates the array with the elements of this collection.\n */\n@sample
samples.collections.Collections.Collections.collectionToTypedArray\n\n@kotlin.internal.InlineOnly\npublic
actual inline fun <T> Collection<T>.toArray(): Array<T> =
copyToArray(this)\n\n@JsName("copyToArray")\n@PublishedApi\ninternal fun <T> copyToArray(collection:
Collection<T>): Array<T> {\n    return if (collection.asDynamic().toArray !== undefined)\n
collection.asDynamic().toArray().unsafeCast<Array<T>>()\n    else\n
copyToArrayImpl(collection).unsafeCast<Array<T>>()\n}\n\n@JsName("copyToArrayImpl")\ninternal actual fun

```

```

copyToArrayImpl(collection: Collection<*>): Array<Any?> {
    val array = emptyArray<Any?>()
    val iterator = collection.iterator()
    while (iterator.hasNext())
        array.asDynamic().push(iterator.next())
    return array
}

@JsName("copyToExistingArrayImpl")
internal actual fun <T> copyToArrayImpl(collection: Collection<*>, array: Array<T>): Array<T> {
    if (array.size < collection.size)
        return copyToArrayImpl(collection).unsafeCast<Array<T>>()
    val iterator = collection.iterator()
    var index = 0
    while (iterator.hasNext()) {
        array[index++] = iterator.next().unsafeCast<T>()
    }
    if (index < array.size)
        array[index] = null.unsafeCast<T>()
    return array
}

/**
 * Returns an immutable list containing only the specified object
 * [element].
 */
public fun <T> listOf(element: T): List<T> =
    arrayOf(element)

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {
    return ArrayList<E>().apply(builderAction).build()
}

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E> {
    checkBuilderCapacity(capacity)
    return ArrayList<E>(capacity).apply(builderAction).build()
}

/**
 * Returns an immutable set containing only the specified object [element].
 */
public fun <T> setOf(element: T): Set<T> =
    hashSetOf(element)

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {
    return LinkedHashSet<E>().apply(builderAction).build()
}

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit): Set<E> {
    return LinkedHashSet<E>(capacity).apply(builderAction).build()
}

/**
 * Returns an immutable map, mapping only the specified key to the
 * specified value.
 */
public fun <K, V> mapOf(pair: Pair<K, V>): Map<K, V> =
    hashMapOf(pair)

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {
    return LinkedHashMap<K, V>().apply(builderAction).build()
}

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal actual inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {
    return LinkedHashMap<K, V>(capacity).apply(builderAction).build()
}

/**
 * Fills the list with the provided [value].
 * Each element in the list gets replaced with the [value].
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.fill(value: T): Unit {
    for (index in 0..lastIndex)
        this[index] = value
}

/**
 * Randomly shuffles elements in this list.
 * See:
 * https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random())

/**
 * Returns a new list with the elements of this list randomly shuffled.
 */
@SinceKotlin("1.2")
public actual fun <T> Iterable<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }

/**
 * Sorts elements in the list in-place according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * @sample samples.collections.Collections.Sorting.sortMutableList
 */
public actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {
    collectionsSort(this, naturalOrder())
}

/**
 * Sorts elements in the list in-place according to the order specified with [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * @sample samples.collections.Collections.Sorting.sortMutableListWith
 */
public actual fun <T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit {
    collectionsSort(this, comparator)
}

private fun <T> collectionsSort(list: MutableList<T>, comparator: Comparator<in T>) {
    if (list.size <= 1) return
    val array = copyToArray(list)
    sortArrayWith(array, comparator)
    for (i in 0 until array.size)
        list[i] = array[i]
}

internal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int): Array<T> {
    return

```

```

arrayOfNulls<Any>(size).unsafeCast<Array<T>>()\n}\n\n@SinceKotlin(\`1.3\`)\n@PublishedApi\n@JsName(\`"arrayCopy"\`)\ninternal fun <T> arrayCopy(source:
    Array<out T>, destination: Array<in T>, destinationOffset: Int, startIndex: Int, endIndex: Int) {\n
AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)\n    val rangeSize = endIndex - startIndex\n
AbstractList.checkRangeIndexes(destinationOffset, destinationOffset + rangeSize, destination.size)\n    if
(js(\`"ArrayBuffer"\`).isView(destination) && js(\`"ArrayBuffer"\`).isView(source)) {\n        val subrange =
source.asDynamic().subarray(startIndex, endIndex)\n        destination.asDynamic().set(subrange,
destinationOffset)\n    } else {\n        if (source !== destination || destinationOffset <= startIndex) {\n            for
(index in 0 until rangeSize) {\n                destination[destinationOffset + index] = source[startIndex + index]\n
            }\n        } else {\n            for (index in rangeSize - 1 downTo 0) {\n                destination[destinationOffset + index] =
source[startIndex + index]\n            }\n        }\n    }\n}\n}\n}\n}\n}\n\n// no singleton
map implementation in js, return map as is\n@Suppress(\`"NOTHING_TO_INLINE"\`)\ninternal actual inline fun
<K, V> Map<K, V>.toSingletonMapOrSelf(): Map<K, V> =
this\n\n@Suppress(\`"NOTHING_TO_INLINE"\`)\ninternal actual inline fun <K, V> Map<out K,
V>.toSingletonMap(): Map<K, V> = this.toMutableMap()\n\n\n@Suppress(\`"NOTHING_TO_INLINE"\`)\ninternal
actual inline fun <T> Array<out T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =\n    if
(isVarargs)\n        // no need to copy vararg array in JS\n        this\n    else\n        this.copyOfOf()\n\n\n\n@PublishedApi\ninternal actual fun checkIndexOverflow(index: Int): Int {\n    if (index < 0)
{\n        throwIndexOverflow()\n    }\n    return index\n}\n\n\n@PublishedApi\ninternal actual fun
checkCountOverflow(count: Int): Int {\n    if (count < 0) {\n        throwCountOverflow()\n    }\n    return
count\n}\n\n\n/**\n * JS map and set implementations do not make use of capacities or load factors.\n
*/\n\n@PublishedApi\ninternal actual fun mapCapacity(expectedSize:
    Int) = expectedSize\n\n\n/**\n * Checks a collection builder function capacity argument.\n * In JS no validation is
made in Map/Set constructor yet.\n */\n\n@SinceKotlin(\`1.3\`)\n@PublishedApi\ninternal fun
checkBuilderCapacity(capacity: Int) {\n    require(capacity >= 0) { \`"capacity must be non-negative."\`
}\n}\n}\n\ninternal actual fun brittleContainsOptimizationEnabled(): Boolean = false", /*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\`"CollectionsKt"\`)\n\n\npackage
kotlin.collections\n\n\n/**\n * Returns the given iterator itself. This allows to use an instance of iterator in a `for`
loop.\n * @sample samples.collections.Iterators.iterator\n */\n\n@kotlin.internal.InlineOnly\npublic inline operator
fun <T> Iterator<T>.iterator(): Iterator<T> =
    this\n\n\n/**\n * Returns an [Iterator] that wraps each element produced by the original iterator\n * into an
[IndexValue] containing the index of that element and the element itself.\n */\n\n * @sample
samples.collections.Iterators.withIndexIterator\n */\n\npublic fun <T> Iterator<T>.withIndex():
    Iterator<IndexedValue<T>> = IndexingIterator(this)\n\n\n/**\n * Performs the given [operation] on each element of
this [Iterator].\n * @sample samples.collections.Iterators.forEachIterator\n */\n\npublic inline fun <T>
    Iterator<T>.forEach(operation: (T) -> Unit): Unit {\n    for (element in this) operation(element)\n}\n\n\n/**\n *
Iterator transforming original `iterator` into iterator of [IndexedValue], counting index from zero.\n */\n\ninternal class
    IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {\n    private var index =
0\n    final override fun hasNext(): Boolean = iterator.hasNext()\n    final override fun next(): IndexedValue<T> =
IndexedValue(checkIndexOverflow(index++),
        iterator.next())\n}\n\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\`"ComparisonsKt"\`)\n\n\npackage
kotlin.comparisons\n\n\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport kotlin.random.*\n\n\n/**\n * Returns the

```

greater of two values.
`* If values are equal, returns the first one.`
`@SinceKotlin("1.1")\npublic expect fun <T : Comparable<T>> maxOf(a: T, b: T): T\n`
`* Returns the greater of two values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Byte, b: Byte): Byte\n`
`* Returns the greater of two values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Short, b: Short): Short\n`
`* Returns the greater of two values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Int, b: Int): Int\n`
`* Returns the greater of two values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Long, b: Long): Long\n`
`* Returns the greater of two values.`
`* If either value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Float, b: Float): Float\n`
`* Returns the greater of two values.`
`* If either value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Double, b: Double): Double\n`
`* Returns the greater of three values.`
`* If there are multiple equal maximal values, returns the first of them.`
`@SinceKotlin("1.1")\npublic expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T\n`
`* Returns the greater of three values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte\n`
`* Returns the greater of three values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Short, b: Short, c: Short): Short\n`
`* Returns the greater of three values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Int, b: Int, c: Int): Int\n`
`* Returns the greater of three values.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Long, b: Long, c: Long): Long\n`
`* Returns the greater of three values.`
`* If any value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Float, b: Float, c: Float): Float\n`
`* Returns the greater of three values.`
`* If any value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Double, b: Double, c: Double): Double\n`
`* Returns the greater of three values according to the order specified by the given [comparator].`
`* If there are multiple equal maximal values, returns the first of them.`
`@SinceKotlin("1.1")\npublic fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n return maxOf(a, maxOf(b, c, comparator), comparator)\n}\n`
`* Returns the greater of two values according to the order specified by the given [comparator].`
`* If values are equal, returns the first one.`
`@SinceKotlin("1.1")\npublic fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T {\n return if (comparator.compare(a, b) >= 0) a else b\n}\n`
`* Returns the greater of the given values.`
`* If there are multiple equal maximal values, returns the first of them.`
`@SinceKotlin("1.4")\npublic expect fun <T : Comparable<T>> maxOf(a: T, vararg other: T): T\n`
`* Returns the greater of the given values.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Byte, vararg other: Byte): Byte\n`
`* Returns the greater of the given values.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Short, vararg other: Short): Short\n`
`* Returns the greater of the given values.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Int, vararg other: Int): Int\n`
`* Returns the greater of the given values.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Long, vararg other: Long): Long\n`
`* Returns the greater of the given values.`
`* If any value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Float, vararg other: Float): Float\n`
`* Returns the greater of the given values.`
`* If any value is `NaN`, returns `NaN`.`
`@SinceKotlin("1.4")\npublic expect fun maxOf(a: Double, vararg other: Double): Double\n`
`* Returns the greater of the given values according to the order specified by the given [comparator].`
`* If there are multiple equal maximal values, returns the first of them.`
`@SinceKotlin("1.4")\npublic fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T {\n var max = a\n for (e in other) if (comparator.compare(max, e) < 0) max = e\n return max\n}\n`
`* Returns the smaller of two values.`
`* If values are equal, returns the`

first one.

```

@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> minOf(a: T, b: T): T
Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte): Byte
Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short): Short
Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int): Int
Returns the smaller of two values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long): Long
Returns the smaller of two values.
If either value is NaN, returns NaN.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float): Float
Returns the smaller of two values.
If either value is NaN, returns NaN.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double): Double
Returns the smaller of two values.
If either value is NaN, returns NaN.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double, c: Double): Double
Returns the smaller of three values.
If there are multiple equal minimal values, returns the first of them.

@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T
Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte
Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short, c: Short): Short
Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int, c: Int): Int
Returns the smaller of three values.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long, c: Long): Long
Returns the smaller of three values.
If any value is NaN, returns NaN.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float, c: Float): Float
Returns the smaller of three values.
If any value is NaN, returns NaN.

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double, c: Double): Double
Returns the smaller of three values according to the order specified by the given [comparator].
If there are multiple equal minimal values, returns the first of them.

@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T
return minOf(a, minOf(b, c, comparator), comparator)
Returns the smaller of two values according to the order specified by the given [comparator].
If values are equal, returns the first one.

@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T
return if (comparator.compare(a, b) <= 0) a else b
Returns the smaller of the given values.
If there are multiple equal minimal values, returns the first of them.

@SinceKotlin("1.4")
public expect fun <T : Comparable<T>> minOf(a: T, vararg other: T): T
Returns the smaller of the given values.

@SinceKotlin("1.4")
public expect fun minOf(a: Byte, vararg other: Byte): Byte
Returns the smaller of the given values.

@SinceKotlin("1.4")
public expect fun minOf(a: Short, vararg other: Short): Short
Returns the smaller of the given values.

@SinceKotlin("1.4")
public expect fun minOf(a: Int, vararg other: Int): Int
Returns the smaller of the given values.

@SinceKotlin("1.4")
public expect fun minOf(a: Long, vararg other: Long): Long
Returns the smaller of the given values.
If any value is NaN, returns NaN.

@SinceKotlin("1.4")
public expect fun minOf(a: Float, vararg other: Float): Float
Returns the smaller of the given values.
If any value is NaN, returns NaN.

@SinceKotlin("1.4")
public expect fun minOf(a: Double, vararg other: Double): Double
Returns the smaller of the given values according to the order specified by the given [comparator].
If there are multiple equal minimal values, returns the first of them.

@SinceKotlin("1.4")
public fun <T> minOf(a: T, vararg other: T, comparator: Comparator<in T>): T
var min = a
for (e in other) if (comparator.compare(min, e) > 0) min = e
return

```

```

min\n}\n\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns the first non-null value produced by
[transform] function being applied to entries of this map in iteration order,\n * or throws
[NoSuchElementException] if no non-null value was produced.\n * \n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K,
V>.firstNotNullOf(transform: (Map.Entry<K, V>) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the map was transformed to a non-null value.")\n}\n\n/**\n * Returns
the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n *
or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K,
V>.firstNotNullOfOrNull(transform: (Map.Entry<K, V>) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n *
Returns a [List]
containing all key-value pairs.\n *\npublic fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>> {\n    if (size
== 0)\n        return emptyList()\n    val iterator = entries.iterator()\n    if (!iterator.hasNext())\n        return
emptyList()\n    val first = iterator.next()\n    if (!iterator.hasNext())\n        return listOf(first.toPair())\n    val result =
ArrayList<Pair<K, V>>(size)\n    result.add(first.toPair())\n    do {\n        result.add(iterator.next().toPair())\n    }
while (iterator.hasNext())\n    return result\n}\n\n/**\n * Returns a single list of all elements yielded from results of
[transform] function being invoked on each entry of original map.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n\npublic inline fun <K, V, R> Map<out K,
V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function
being invoked on each entry of original map.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <K, V, R> Map<out K,
V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
entry of original map, to the given [destination].\n *\npublic inline fun <K, V, R, C : MutableCollection<in R>>
Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Iterable<R>): C {\n    for (element in
this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n *
Appends all elements yielded from results of [transform] function being invoked on each entry of original
map, to the given [destination].\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequenceTo")\npublic inline fun <K, V, R, C :
MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) ->
Sequence<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n
    }\n    return destination\n}\n\n/**\n * Returns a list containing the results of applying the given [transform]
function\n * to each entry in the original map.\n * \n * @sample
samples.collections.Collections.Transformations.mapToList\n
*\n\npublic inline fun <K, V, R> Map<out K,
V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {\n    return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform]

```

```

function\n * to each entry in the original map.\n * \n * @sample
samples.collections.Maps.Transformations.mapNotNull\n * \n\npublic inline fun <K, V, R : Any> Map<out K,
V>.mapNotNull(transform: (Map.Entry<K, V>) -> R?): List<R> {\n    return mapNotNullTo(ArrayList<R>()),
transform)\n}\n\n/**\n * Applies the given [transform] function to each entry in the original map\n * and appends
only the non-null results to the given [destination].\n * \n\npublic inline fun <K, V, R : Any, C : MutableCollection<in
R>> Map<out K, V>.mapNotNullTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {\n    forEach {
element -> transform(element)?.let { destination.add(it) } }\n    return destination\n}\n\n/**\n * Applies the given
[transform] function to each entry of the original map\n * and appends the results to the given [destination].\n
*\n\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.mapTo(destination: C, transform:
(Map.Entry<K, V>) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return
destination\n}\n\n/**\n
* Returns `true` if all entries match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n * \n\npublic inline fun <K, V> Map<out K, V>.all(predicate:
(Map.Entry<K, V>) -> Boolean): Boolean {\n    if (isEmpty()) return true\n    for (element in this) if
(!predicate(element)) return false\n    return true\n}\n\n/**\n * Returns `true` if map has at least one entry.\n * \n *
@sample samples.collections.Collections.Aggregates.any\n * \n\npublic fun <K, V> Map<out K, V>.any(): Boolean
{\n    return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one entry matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun <K, V> Map<out K,
V>.any(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n    if (isEmpty()) return false\n    for (element in
this) if (predicate(element)) return true\n    return false\n}\n\n/**\n * Returns the number of entries in this map.\n
*\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun <K, V> Map<out K, V>.count(): Int {\n    return size\n}\n\n/**\n * Returns the number of entries
matching the given [predicate].\n * \n\npublic inline fun <K, V> Map<out K, V>.count(predicate: (Map.Entry<K, V>)
-> Boolean): Int {\n    if (isEmpty()) return 0\n    var count = 0\n    for (element in this) if (predicate(element))
++count\n    return count\n}\n\n/**\n * Performs the given [action] on each entry.\n
*\n\n@kotlin.internal.HidesMembers\n\npublic inline fun <K, V> Map<out K, V>.forEach(action: (Map.Entry<K, V>)
-> Unit): Unit {\n    for (element in this) action(element)\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V, R : Comparable<R>>
Map<out K, V>.maxBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return
maxByOrNull(selector)\n}\n\n/**\n
* Returns the first entry yielding the largest value of the given function or `null` if there are no entries.\n * \n *
@sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V, R : Comparable<R>> Map<out
K, V>.maxByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return
entries.maxByOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V>) -> Double): Double {\n    return entries.maxOf(selector)\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V>) -> Float): Float {\n    return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * @throws

```

NoSuchElementException if the map is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,\nV>.maxOf(selector: (Map.Entry<K, V>) -> R):
```

```
R {\n    return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value among all values produced by\n [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values\n produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,\nV>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n    return\nentries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]\n function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by\n [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\ninline fun <K, V> Map<out K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n    return\nentries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]\n function\n * applied to each entry in the map or `null` if there are no entries.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,\nV>.maxOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n    return entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\ninline fun <K, V, R> Map<out K, V>.maxOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) ->\nR): R {\n    return entries.maxOfWith(comparator, selector)\n}\n\n/**\n * Returns the largest value according to the\n provided [comparator]\n * among all values produced by [selector] function applied to each entry in the map or\n `null` if there are no entries.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,\nV>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n    return\nentries.maxOfWithOrNull(comparator, selector)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,\nV>.maxWith(comparator:
```

```
Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n    return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided [comparator] or `null` if there are no\n entries.\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,\nV>.maxWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n    return\nentries.maxWithOrNull(comparator)\n}\n\n@Deprecated("Use minByOrNull instead.")\nReplaceWith("this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =\n"1.5", hiddenSince = "1.6")\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minBy(selector:\n(Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first\n entry yielding the smallest value of the given function or `null` if there are no entries.\n * \n * @sample\n samples.collections.Collections.Aggregates.minByOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out\nK, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return
```


`entries.minByOrNull(selector)` Returns the smallest value among all values produced by [selector] function applied to each entry in the map. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the map is empty.

```

*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Double): Double {
    return entries.minOf(selector)
}
*\/ Returns the smallest
value among all values produced by [selector] function applied to each entry in the map. If any of values
produced by [selector] function is NaN, the
returned result is NaN. @throws NoSuchElementException if the map is empty.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Float): Float {
    return entries.minOf(selector)
}
*\/ Returns the smallest value
among all values produced by [selector] function applied to each entry in the map. @throws
NoSuchElementException if the map is empty.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOf(selector: (Map.Entry<K, V>) -> R): R {
    return entries.minOf(selector)
}
*\/ Returns the
smallest value among all values produced by [selector] function applied to each entry in the map or null if
there are no entries. If any of values produced by [selector] function is NaN, the returned result is
NaN.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {
    return
entries.minOfOrNull(selector)
}
*\/ Returns the smallest value among all values produced by [selector]
function applied to each entry in the map or null if there are no entries. If any of values produced by
[selector] function is NaN, the returned result is NaN.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {
    return
entries.minOfOrNull(selector)
}
*\/ Returns
the smallest value among all values produced by [selector] function applied to each entry in the map or null if
there are no entries.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {
    return entries.minOfOrNull(selector)
}
*\/ Returns
the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each entry in the map. @throws NoSuchElementException if the map is empty.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.minOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {
    return
entries.minOfWith(comparator,
selector)
}
*\/ Returns the smallest value according to the provided [comparator] among all values
produced by [selector] function applied to each entry in the map or null if there are no entries.
*\/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.minOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {
    return
entries.minOfWithOrNull(comparator, selector)
}
@Deprecated("Use minWithOrNull instead.",
ReplaceWith("this.minWithOrNull(comparator)"))@DeprecatedSinceKotlin(warningSince = "1.4", errorSince

```

```

= "1.5", hiddenSince = "1.6")\npublic fun <K, V> Map<out K, V>.minWith(comparator: Comparator<in
Map.Entry<K, V>>): Map.Entry<K, V>? {\n    return minWithOrNull(comparator)\n}\n\n/**\n * Returns the first
entry having the smallest value according to the provided [comparator]
or `null` if there are no entries.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
Map<out K, V>.minWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n    return
entries.minWithOrNull(comparator)\n}\n\n/**\n * Returns `true` if the map has no entries.\n */\n * @sample
samples.collections.Collections.Aggregates.none\n */\npublic fun <K, V> Map<out K, V>.none(): Boolean {\n
return isEmpty()\n}\n\n/**\n * Returns `true` if no entries match the given [predicate].\n */\n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <K, V> Map<out K,
V>.none(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n    if (isEmpty()) return true\n    for (element in
this) if (predicate(element)) return false\n    return true\n}\n\n/**\n * Performs the given [action] on each entry and
returns the map itself afterwards.\n */\n@SinceKotlin("1.1")\npublic inline fun <K, V, M : Map<out K, V>>
M.onEach(action:
(Map.Entry<K, V>) -> Unit): M {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs
the given [action] on each entry, providing sequential index with the entry,\n * and returns the map itself
afterwards.\n * @param [action] function that takes the index of an entry and the entry itself\n * and performs the
action on the entry.\n */\n@SinceKotlin("1.4")\npublic inline fun <K, V, M : Map<out K, V>>
M.onEachIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): M {\n    return apply {
entries.forEachIndexed(action) }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original map returning
its entries when being iterated.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.asIterable(): Iterable<Map.Entry<K, V>> {\n    return entries\n}\n\n/**\n * Creates a [Sequence] instance that
wraps the original map returning its entries when being iterated.\n */\npublic fun <K, V> Map<out K,
V>.asSequence(): Sequence<Map.Entry<K,
V>> {\n    return entries.asSequence()\n}\n\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\npackage kotlin.text\n\n// NOTE: THIS FILE IS AUTO-
GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n// 10 mappings totally\ninternal fun
Char.titlecaseImpl(): String {\n    val uppercase = uppercase()\n    if (uppercase.length > 1) {\n        return if (this ==
"\u0149") uppercase else uppercase[0] + uppercase.substring(1).lowercase()\n    }\n    return
titlecaseChar().toString()\n}\n\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin.text\n\n/**\n * Converts this character
to lower case using Unicode mapping rules of the invariant locale.\n */\n@Deprecated("Use lowercaseChar()
instead.", ReplaceWith("lowercaseChar()"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toLowerCase(): Char =
lowercaseChar()\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant
locale.\n * This function performs one-to-one character mapping.\n * To support one-to-many character
mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself is
returned.\n */\n * @sample samples.text.Chars.lowercase\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c actual inline fun Char.lowercaseChar(): Char = lowercase()[0]\n\n/**\n * Converts this character to lower case
using Unicode mapping rules of the invariant locale.\n * This function supports one-to-many character mapping,
thus the length of the returned string can be greater than one.\n * For example, `'\u0130'.lowercase()` returns
`'\u0069\u0307'`,\n * where `'\u0130` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character
(`\u0130`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n
*/\n * @sample samples.text.Chars.lowercase\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli

```

```

c actual inline fun Char.lowercase(): String = toString().asDynamic().toLowerCase().unsafeCast<String>()\n\n**\n
* Converts this character to upper case using Unicode mapping rules of the invariant locale.\n
*\n@Deprecated("Use uppercaseChar() instead.")\n
ReplaceWith("uppercaseChar()")\n@DeprecatedSinceKotlin(warningSince =
`1.5`)\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toUpperCase(): Char =
uppercaseChar()\n\n**\n * Converts this character to upper case using Unicode mapping rules
of the invariant locale.\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself
is returned.\n * @sample samples.text.Chars.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Char.uppercaseChar(): Char {\n    val uppercase = uppercase()\n    return if (uppercase.length > 1) this else
uppercase[0]\n}\n\n**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n * This function supports one-to-many character mapping, thus the length of the returned string can be
greater than one.\n * For example, `'\uFB00'.uppercase()` returns `'\u0046\u0046'`,\n * where `'\uFB00` is the
LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no upper case mapping, the result of
`toString()` of this char is returned.\n * @sample
samples.text.Chars.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c actual inline fun Char.uppercase(): String = toString().asDynamic().toUpperCase().unsafeCast<String>()\n\n**\n
* Converts this character to title case using Unicode mapping rules of the invariant locale.\n * This function
performs one-to-one character mapping.\n * To support one-to-many character mapping use the [titlecase]
function.\n * If this character has no mapping equivalent, the result of calling [uppercaseChar] is returned.\n * @
sample samples.text.Chars.titlecase\n
*\n@SinceKotlin("1.5")\npublic actual fun Char.titlecaseChar(): Char =
titlecaseCharImpl()\n\n**\n * Returns `true` if this character is a Unicode high-surrogate code unit (also known as
leading-surrogate code unit).\n *\npublic actual fun Char.isHighSurrogate(): Boolean = this in
Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE\n\n**\n * Returns `true` if this
character is a Unicode low-surrogate code unit (also known as trailing-surrogate code unit).\n *\npublic actual fun
Char.isLowSurrogate(): Boolean = this in
Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE\n\n**\n * Returns the Unicode general
category of this character.\n *\n@SinceKotlin("1.5")\npublic actual val Char.category: CharCategory\n    get() =
CharCategory.valueOf(getCategoryValue())\n\n**\n * Returns `true` if this character (Unicode code point) is
defined in Unicode.\n * A character is considered to be defined in Unicode if its [category] is not
[CharCategory.UNASSIGNED].\n *\n@SinceKotlin("1.5")\npublic actual fun Char.isDefined(): Boolean {\n    if
(this < "\u0080") {\n        return true\n    }\n    return getCategoryValue() !=
CharCategory.UNASSIGNED.value\n}\n\n**\n * Returns `true` if this character is a letter.\n * A character is
considered to be a letter if its [category] is [CharCategory.UPPERCASE_LETTER],\n *
[CharCategory.LOWERCASE_LETTER],
[CharCategory.TITLECASE_LETTER], [CharCategory.MODIFIER_LETTER], or
[CharCategory.OTHER_LETTER].\n * @sample samples.text.Chars.isLetter\n
*\n@SinceKotlin("1.5")\npublic actual fun Char.isLetter(): Boolean {\n    if (this in 'a'..'z' || this in 'A'..'Z') {\n
return true\n    }\n    if (this < "\u0080") {\n        return false\n    }\n    return isLetterImpl()\n}\n\n**\n * Returns
`true` if this character is a letter or digit.\n * @see isLetter\n * @see isDigit\n * @sample
samples.text.Chars.isLetterOrDigit\n
*\n@SinceKotlin("1.5")\npublic actual fun Char.isLetterOrDigit(): Boolean
{\n    if (this in 'a'..'z' || this in 'A'..'Z' || this in '0'..'9') {\n        return true\n    }\n    if (this <
"\u0080") {\n        return
false\n    }\n    return isDigitImpl() || isLetterImpl()\n}\n\n**\n * Returns `true` if this character is a digit.\n *
A character is considered to be a digit if its [category] is [CharCategory.DECIMAL_DIGIT_NUMBER].\n * @sample
samples.text.Chars.isDigit\n

```

```

*/\n@SinceKotlin("1.5")\npublic actual fun Char.isDigit(): Boolean {\n    if (this in '0'..'9') {\n        return true\n    }\n    if (this < "\u0080") {\n        return false\n    }\n    return isDigitImpl()\n}\n\n/**\n * Returns `true` if this character is upper case.\n */\n * A character is considered to be an upper case character if its [category] is [CharCategory.UPPERCASE_LETTER],\n * or it has contributory property `Other_Uppercase` as defined by the Unicode Standard.\n */\n * @sample samples.text.Chars.isUpperCase\n*/\n@SinceKotlin("1.5")\npublic actual fun Char.isUpperCase(): Boolean {\n    if (this in 'A'..'Z') {\n        return true\n    }\n    if (this < "\u0080") {\n        return false\n    }\n    return isUpperCaseImpl()\n}\n\n/**\n * Returns `true` if this character is lower case.\n */\n * A character is considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has contributory property `Other_Lowercase` as defined by the Unicode Standard.\n */\n * @sample samples.text.Chars.isLowerCase\n*/\n@SinceKotlin("1.5")\npublic actual fun Char.isLowerCase(): Boolean {\n    if (this in 'a'..'z') {\n        return true\n    }\n    if (this < "\u0080") {\n        return false\n    }\n    return isLowerCaseImpl()\n}\n\n/**\n * Returns `true` if this character is a title case letter.\n */\n * A character is considered to be a title case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n */\n * @sample samples.text.Chars.isTitleCase\n*/\n@SinceKotlin("1.5")\npublic actual fun Char.isTitleCase(): Boolean {\n    if (this < "\u0080") {\n        return false\n    }\n    return getCategoryValue() == CharCategory.TITLECASE_LETTER.value\n}\n\n/**\n * Returns `true` if this character is an ISO control character.\n */\n * A character is considered to be an ISO control character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range "\u0000".." \u001F" or in the range "\u007F".." \u009F".\n */\n * @sample samples.text.Chars.isISOControl\n*/\n@SinceKotlin("1.5")\npublic actual fun Char.isISOControl(): Boolean {\n    return this <= "\u001F" || this in "\u007F".." \u009F"\n}\n\n/**\n * Determines whether a character is whitespace according to the Unicode standard.\n */\n * Returns `true` if the character is whitespace.\n */\n * @sample samples.text.Chars.isWhitespace\n*/\npublic actual fun Char.isWhitespace(): Boolean = isWhitespaceImpl(), /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport kotlin.js.RegExp\n\n/**\n * Converts the characters in the specified array to a string.\n */\n@SinceKotlin("1.2")\n@Deprecated("Use CharArray.concatToString() instead", ReplaceWith("chars.concatToString()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5")\npublic actual fun String(chars: CharArray): String {\n    var result = ""\n    for (char in chars) {\n        result += char\n    }\n    return result\n}\n\n/**\n * Converts the characters from a portion of the specified array to a string.\n */\n * @throws IndexOutOfBoundsException if either [offset] or [length] are less than zero\n * or `offset + length` is out of [chars] array bounds.\n */\n@SinceKotlin("1.2")\n@Deprecated("Use CharArray.concatToString(startIndex, endIndex) instead", ReplaceWith("chars.concatToString(offset, offset + length)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5")\npublic actual fun String(chars: CharArray, offset: Int, length: Int): String {\n    if (offset < 0 || length < 0 || chars.size - offset < length)\n        throw IndexOutOfBoundsException("size: ${chars.size}; offset: $offset; length: $length")\n    var result = ""\n    for (index in offset until offset + length) {\n        result += chars[index]\n    }\n    return result\n}\n\n/**\n * Concatenates characters in this [CharArray] into a String.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun CharArray.concatToString(): String {\n    var result = ""\n    for (char in this) {\n        result += char\n    }\n    return result\n}\n\n/**\n * Concatenates characters in this [CharArray] or its subrange into a String.\n */\n * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.\n * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.\n */\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun CharArray.concatToString(startIndex:

```



```

asDynamic().toUpperCase()\n\n**\n * Returns a copy of this string converted to upper case using Unicode mapping
rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character mapping,\n *
thus the length of the
returned string can be different from the length of the original string.\n *\n * @sample
samples.text.Strings.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c actual inline fun String.toUpperCase(): String = asDynamic().toUpperCase()\n\n**\n * Returns a copy of this string
converted to lower case using the rules of the default locale.\n *\n@Deprecated("Use lowercase() instead."),
ReplaceWith("lowercase()")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String =
asDynamic().toLowerCase()\n\n**\n * Returns a copy of this string converted to lower case using Unicode
mapping rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character
mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n *
@sample samples.text.Strings.lowercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c
actual inline fun String.toLowerCase(): String = asDynamic().toLowerCase()\n\n@kotlin.internal.InlineOnly\ninternal
actual inline fun String.nativeIndexOf(str: String, fromIndex: Int): Int = asDynamic().indexOf(str,
fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeLastIndexOf(str: String,
fromIndex: Int): Int = asDynamic().lastIndexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal inline fun
String.nativeStartsWith(s: String, position: Int): Boolean = asDynamic().startsWith(s,
position)\n\n@kotlin.internal.InlineOnly\ninternal inline fun String.nativeEndsWith(s: String): Boolean =
asDynamic().endsWith(s)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun String.substring(startIndex: Int):
String = asDynamic().substring(startIndex)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun
String.substring(startIndex: Int, endIndex: Int):
String = asDynamic().substring(startIndex, endIndex)\n\n@Deprecated("Use String.plus() instead",
ReplaceWith("this + str"))\n@DeprecatedSinceKotlin(warningSince =
"1.6")\n@kotlin.internal.InlineOnly\npublic inline fun String.concat(str: String): String =
asDynamic().concat(str)\n\n@Deprecated("Use Regex.findAll() instead or invoke matches() on String dynamically:
this.asDynamic().match(regex)")\n@DeprecatedSinceKotlin(warningSince =
"1.6")\n@kotlin.internal.InlineOnly\npublic inline fun String.match(regex: String): Array<String>? =
asDynamic().match(regex)\n\n//native public fun String.trim(): String\n//TODO: String.replace to implement
effective trimLeading and trimTrailing\n\n@kotlin.internal.InlineOnly\ninternal inline fun
String.nativeReplace(pattern: RegExp, replacement: String): String = asDynamic().replace(pattern,
replacement)\n\n**\n * Compares two strings lexicographically, optionally ignoring case differences.\n *\n * If
[ignoreCase] is true, the result of `Char.toUpperCaseChar().toLowerCaseChar()`
on each character is compared.\n
*\n@SinceKotlin("1.2")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun String.compareTo(other: String, ignoreCase: Boolean = false): Int {\n    if (ignoreCase) {\n        val n1 =
this.length\n        val n2 = other.length\n        val min = minOf(n1, n2)\n        if (min == 0) return n1 - n2\n        for
(index in 0 until min) {\n            var thisChar = this[index]\n            var otherChar = other[index]\n            if
(thisChar != otherChar) {\n                thisChar = thisChar.toUpperCaseChar()\n                otherChar =
otherChar.toUpperCaseChar()\n                if (thisChar != otherChar) {\n                    thisChar =
thisChar.toLowerCaseChar()\n                    otherChar = otherChar.toLowerCaseChar()\n                    if (thisChar !=
otherChar) {\n                        return thisChar.compareTo(otherChar)\n                    }\n                }\n            }\n            return n1 - n2\n        } else {\n            return compareTo(other)\n        }\n    }\n\n**\n * Returns `true` if the
contents of this char sequence are equal to the contents of the specified [other],\n * i.e. both char sequences contain
the same number of the same characters in the same order.\n *\n * @sample samples.text.Strings.contentEquals\n

```

```

*\/n@SinceKotlin("1.5")\npublic actual infix fun CharSequence?.contentEquals(other: CharSequence?): Boolean =
contentEqualsImpl(other)\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of
the specified [other], optionally ignoring case difference.\n *\/n * @param ignoreCase `true` to ignore character case
when comparing contents.\n *\/n * @sample samples.text.Strings.contentEquals\n *\/n@SinceKotlin("1.5")\npublic
actual fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean {\n    return if
(ignoreCase)\n        this.contentEqualsIgnoreCaseImpl(other)\n    else\n        this.contentEqualsImpl(other)\n}\n\nprivate val STRING_CASE_INSENSITIVE_ORDER =
Comparator<String> { a, b -> a.compareTo(b, ignoreCase = true) }\n\n@SinceKotlin("1.2")\npublic actual val
String.Companion.CASE_INSENSITIVE_ORDER: Comparator<String>\n    get() =
STRING_CASE_INSENSITIVE_ORDER\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n

```

```

*\/n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CharsKt")\n\npackage kotlin.text\n\n/**\n * Returns the numeric value of the decimal digit that this Char represents.\n * Throws an exception if this Char is
not a valid decimal digit.\n *\/n * A Char is considered to represent a decimal digit if [isDigit] is true for the Char.\n
* In this case, the Unicode decimal digit value of the character is returned.\n *\/n * @sample
samples.text.Chars.digitToInt\n

```

```

*\/n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
fun Char.digitToInt(): Int {\n    return digitOf(this, 10).also {\n        if (it < 0) throw
IllegalArgumentException("Char $this is not a decimal digit")\n    }\n}\n\n/**\n * Returns the numeric value of the
digit that this Char represents in the specified [radix].\n * Throws an exception if the [radix] is not in the range
`2..36` or if this Char is not a valid digit in the specified [radix].\n *\/n * A Char is considered to represent a digit in
the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the Char and the Unicode
decimal digit value of the character is less than the specified [radix]. In this case the decimal digit value is
returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix +
'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the lowercase Latin
letters 'a' through
'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.\n * - The
Char is one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and its [code] is less than `radix +
0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth Latin small
letters '\uFF41' through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41
+ 10` is returned.\n *\/n * @sample samples.text.Chars.digitToInt\n

```

```

*\/n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToInt(radix:
Int): Int {\n    return digitToIntOrNull(radix) ?: throw IllegalArgumentException("Char $this is not a digit in the
given radix=$radix")\n}\n\n/**\n * Returns the numeric value of the decimal digit that this Char represents, or
`null` if this Char is not a valid decimal digit.\n *\/n * A Char is considered to represent a decimal digit
if [isDigit] is true for the Char.\n * In this case, the Unicode decimal digit value of the character is returned.\n *\/n *
@sample samples.text.Chars.digitToIntOrNull\n

```

```

*\/n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
Char.digitToIntOrNull(): Int? {\n    return digitOf(this, 10).takeIf { it >= 0 }\n}\n\n/**\n * Returns the numeric
value of the digit that this Char represents in the specified [radix], or `null` if this Char is not a valid digit in the
specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36`.\n *\/n * A Char is considered to
represent a digit in the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the Char
and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit
value is returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix
+ 'A'.code - 10`. In
this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the lowercase Latin letters 'a' through 'z'
and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.\n * - The Char is

```

one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth Latin small letters '\uFF41' through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.\n * \n * @sample samples.text.Chars.digitToIntOrNull\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToIntOrNull(radix: Int): Int? {\n    checkRadix(radix)\n    return digitOf(this, radix).takeIf { it >= 0 }\n}\n\n/**\n * Returns the Char that represents this decimal digit.\n * Throws an exception if this value is not in the range
```

```
`0..9`. \n * \n * If this value is in `0..9`, the decimal digit Char with code `0'.code + this` is returned.\n * \n * @sample samples.text.Chars.digitToChar\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(): Char {\n    if (this in 0..9) {\n        return '0' + this\n    }\n    throw IllegalArgumentException("Int $this is not a decimal digit")\n}\n\n/**\n * Returns the Char that represents this numeric digit value in the specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36` or if this value is not in the range `0 until radix`. \n * \n * If this value is less than `10`, the decimal digit Char with code `0'.code + this` is returned.\n * Otherwise, the uppercase Latin letter with code `A'.code + this - 10` is returned.\n * \n * @sample samples.text.Chars.digitToChar\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(radix: Int): Char {\n    if (radix !in 2..36) {\n        throw IllegalArgumentException("Invalid radix: $radix. Valid radix values are in range 2..36")\n    }\n    if (this < 0 || this >= radix) {\n        throw IllegalArgumentException("Digit $this does not represent a valid digit in radix $radix")\n    }\n    return if (this < 10) {\n        '0' + this\n    } else {\n        'A' + this - 10\n    }\n}\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n
```

```
*\n@Deprecated("Use lowercaseChar() instead.")
```

```
ReplaceWith("lowercaseChar()")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun Char.toLowerCase(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n * \n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n * \n * @sample samples.text.Chars.lowercase\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

```
Char.lowercaseChar(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n * \n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\u0130'.lowercase()` returns `'\u0069\u0307'`, where `'\u0130` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (`\ufffd\uuffd`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n * \n * @sample samples.text.Chars.lowercase\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

```
Char.lowercase(): String\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n * \n * @sample samples.text.Chars.lowercase\n
```

```
ReplaceWith("uppercaseChar()")\n@DeprecatedSinceKotlin(warningSince
```

```
= "1.5")\npublic expect fun Char.toUpperCase(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n * \n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n * \n * @sample samples.text.Chars.uppercase\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

```
Char.uppercaseChar(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n * \n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\uFB00'.uppercase()` returns `'\u0046\u0046'`, where `'\uFB00` is the LATIN SMALL LIGATURE FF character (`\ufffd\uuffd\uuffd`).\n * If this character has no upper
```


case mapping,

```
the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun\nChar.uppercase(): String\n\n/**\n * Converts this character to title case using Unicode mapping rules of the\n invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many\n character mapping use the [titlecase] function.\n * If this character has no mapping equivalent, the result of calling\n [uppercaseChar] is returned.\n *\n * @sample samples.text.Chars.titlecase\n *\n@SinceKotlin("1.5")\npublic\nexpect fun Char.titlecaseChar(): Char\n\n/**\n * Converts this character to title case using Unicode mapping rules of the\n invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned\n string can be greater than one.\n * For example, `'\u0046\u0066'.titlecase()` returns `'\u0046\u0066'`,\n * where `'\u0046`\n is the LATIN SMALL LIGATURE FF character (`\ufffd\u0046\u0066`).\n *\n * If this character has no title case\n mapping, the result of [uppercase] is returned instead.\n *\n * @sample samples.text.Chars.titlecase\n *\n@SinceKotlin("1.5")\npublic fun Char.titlecase(): String = titlecaseImpl()\n\n/**\n * Concatenates this Char\n and a String.\n *\n * @sample samples.text.Chars.plus\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun\nChar.plus(other: String): String = this.toString() + other\n\n/**\n * Returns `true` if this character is equal to the\n [other] character, optionally ignoring character case.\n *\n * Two characters are considered equal ignoring case if\n `Char.uppercaseChar().lowercaseChar()` on each character produces the same result.\n *\n * @param ignoreCase\n `true` to ignore character case when comparing characters. By default `false`.\n *\n * @sample\n samples.text.Chars.equals\n *\npublic fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {\n    if\n (this ==\n other) return true\n    if (!ignoreCase) return false\n\n    val thisUpper = this.uppercaseChar()\n    val otherUpper =\n other.uppercaseChar()\n\n    return thisUpper == otherUpper || thisUpper.lowercaseChar() ==\n otherUpper.lowercaseChar()\n}\n\n/**\n * Returns `true` if this character is a Unicode surrogate code unit.\n *\n@public fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE\n\n/**\n * Returns the Unicode general category of this character.\n *\n@SinceKotlin("1.5")\npublic expect val\nChar.category: CharCategory\n\n/**\n * Returns `true` if this character (Unicode code point) is defined in\n Unicode.\n *\n * A character is considered to be defined in Unicode if its [category] is not\n [CharCategory.UNASSIGNED].\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isDefined():\nBoolean\n\n/**\n * Returns `true` if this character is a letter.\n *\n * A character is considered to be a letter if its\n [category] is [CharCategory.UPPERCASE_LETTER],\n * [CharCategory.LOWERCASE_LETTER],\n * [CharCategory.TITLECASE_LETTER], [CharCategory.MODIFIER_LETTER], or\n [CharCategory.OTHER_LETTER].\n *\n * @sample samples.text.Chars.isLetter\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isLetter(): Boolean\n\n/**\n * Returns `true` if this character is a\n letter or digit.\n *\n * @see isLetter\n * @see isDigit\n *\n * @sample samples.text.Chars.isLetterOrDigit\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isLetterOrDigit(): Boolean\n\n/**\n * Returns `true` if this\n character is a digit.\n *\n * A character is considered to be a digit if its [category] is\n [CharCategory.DECIMAL_DIGIT_NUMBER].\n *\n * @sample samples.text.Chars.isDigit\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isDigit(): Boolean\n\n/**\n * Returns `true` if this character is\n upper case.\n *\n * A character is considered to be an upper case character if its [category] is\n [CharCategory.UPPERCASE_LETTER],\n * or it has contributory property `Other_Uppercase` as defined by the\n Unicode Standard.\n *\n * @sample samples.text.Chars.isUpperCase\n *\n@SinceKotlin("1.5")\npublic expect\nfun Char.isUpperCase(): Boolean\n\n/**\n * Returns `true` if this character is lower case.\n *\n * A character is\n considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has\n contributory property `Other_Lowercase` as defined by the Unicode Standard.\n *\n * @sample\n samples.text.Chars.isLowerCase\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isLowerCase():\nBoolean\n\n/**\n * Returns `true` if this character is a title case letter.\n *\n * A character is considered to be a title\n case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n *\n * @sample
```

```

samples.text.Chars.isTitleCase\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isTitleCase(): Boolean\n\n/**\n * Returns `true` if this character is an ISO control character.\n *\n * A character is considered to be an ISO control character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range ``\u0000'..' \u001F`` or in the range ``\u007F'..' \u009F``.\n *\n * @sample
samples.text.Chars.isISOControl\n *\n@SinceKotlin("1.5")\npublic expect fun Char.isISOControl(): Boolean\n\n/**\n * Determines whether a character is whitespace according to the Unicode standard.\n * Returns `true` if the character is whitespace.\n *\n * @sample samples.text.Chars.isWhitespace\n *\npublic expect fun Char.isWhitespace(): Boolean\n"/>\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin\n\n/**\n * Creates a Char with the specified [code], or throws an exception if the [code] is out of `Char.MIN_VALUE.code`..`Char.MAX_VALUE.code`. \n *\n * If the program that calls this function is written in a way that only valid [code] is passed as the argument,\n * using the overload that takes a [UShort] argument is preferable (`Char(intValue.toUShort())`).\n * That overload doesn't check validity of the argument, and may improve program performance when the function is called routinely inside a loop.\n *\n * @sample samples.text.Chars.charFromCode\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Char(code: Int): Char {\n    if (code < Char.MIN_VALUE.code || code > Char.MAX_VALUE.code) {\n        throw IllegalArgumentException("Invalid Char code: $code")\n    }\n    return code.toChar()\n}\n\n/**\n * Creates a Char with the specified [code].\n *\n * @sample samples.text.Chars.charFromCode\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun Char(code: UShort): Char\n\n/**\n * Returns the code of this Char.\n *\n * Code of a Char is the value it was constructed with, and the UTF-16 code unit corresponding to this Char.\n *\n * @sample samples.text.Chars.code\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress("DEPRECATION")\npublic inline val Char.code: Int get() = this.toInt()\n"/>\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\npackage kotlin.sequences\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns `true` if [element] is found in the sequence.\n *\n * The operation is _terminal_.\n *\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n *\npublic fun <T> Sequence<T>.elementAt(index: Int): T {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("Sequence doesn't contain element at index $index.") }\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Elements.elementAtOrElse\n *\npublic fun <T> Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (index < 0)\n        return defaultValue(index)\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    }\n    return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Elements.elementAtOrNull\n *\npublic fun <T> Sequence<T>.elementAtOrNull(index: Int): T? {\n    if (index < 0)\n        return null\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n
```

```

return element\n } \n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null`
if no such element was found.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.find(predicate: (T) -> Boolean): T? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the last
element matching the
given [predicate], or `null` if no such element was found.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns first
element.\n *\n * @throws [NoSuchElementException] if the sequence is empty.\n *\n * The operation is _terminal_.\n
*\n */\npublic fun <T> Sequence<T>.first(): T {\n val iterator = iterator()\n if (!iterator.hasNext())\n throw
NoSuchElementException("Sequence is empty.")\n return iterator.next()\n}\n\n/**\n * Returns the first element
matching the given [predicate].\n *\n * @throws [NoSuchElementException] if no such element is found.\n *\n * The
operation is _terminal_.\n *\n */\npublic inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T {\n for
(element in this) if (predicate(element)) return element\n throw NoSuchElementException("Sequence
contains no element matching the predicate.")\n}\n\n/**\n * Returns the first non-null value produced by
[transform] function being applied to elements of this sequence in iteration order,\n * or throws
[NoSuchElementException] if no non-null value was produced.\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the sequence was transformed to a non-null value.")\n}\n\n/**\n *
Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or `null` if no non-null value was produced.\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Any> Sequence<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this)
{\n val result = transform(element)\n if (result != null) {\n return result\n }\n }\n return
null\n}\n\n/**\n * Returns the first element, or `null` if the sequence is empty.\n *\n * The operation is _terminal_.\n
*\n */\npublic fun <T> Sequence<T>.firstOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext())\n
return null\n return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null`
if element was not found.\n *\n * The operation is _terminal_.\n *\n */\npublic inline fun <T>
Sequence<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return
element\n return null\n}\n\n/**\n * Returns first index of [element], or -1 if the sequence does not contain
element.\n *\n * The operation
is _terminal_.\n *\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {\n
var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (element == item)\n return
index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given
[predicate], or -1 if the sequence does not contain such element.\n *\n * The operation is _terminal_.\n *\n
*/\npublic inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n var index = 0\n for (item in
this) {\n checkIndexOverflow(index)\n if (predicate(item))\n return index\n index++\n }\n return -
1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the sequence does not
contain such element.\n *\n * The operation is _terminal_.\n *\n */\npublic inline fun <T>
Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n var lastIndex = -1\n var index = 0\n for (item in
this) {\n checkIndexOverflow(index)\n if (predicate(item))\n lastIndex = index\n index++\n }\n return
lastIndex\n}\n\n/**\n * Returns the last element.\n *\n * The operation is _terminal_.\n *\n * @throws
NoSuchElementException if the sequence is empty.\n *\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Sequence<T>.last(): T {\n val iterator =
iterator()\n if (!iterator.hasNext())\n throw NoSuchElementException("Sequence is empty.")\n var last =

```

```

iterator.next()\n while (iterator.hasNext())\n     last = iterator.next()\n return last\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n *\n * The operation is _terminal_.\n *\n * @throws
NoSuchElementException if no such element is found.\n *\n * @sample
samples.collections.Collections.Elements.last\n *\npublic inline fun <T> Sequence<T>.last(predicate: (T) ->
Boolean): T {\n     var last:
T? = null\n     var found = false\n     for (element in this) {\n         if (predicate(element)) {\n             last = element\n
found = true\n         }\n     }\n     if (!found) throw NoSuchElementException("Sequence contains no element
matching the predicate.")\n     @Suppress("UNCHECKED_CAST")\n     return last as T\n}\n\n/**\n * Returns last
index of [element], or -1 if the sequence does not contain element.\n *\n * The operation is _terminal_.\n *\npublic
fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {\n     var lastIndex = -1\n
var index = 0\n     for (item in this) {\n         checkIndexOverflow(index)\n         if (element == item)\n
lastIndex = index\n         index++\n     }\n     return lastIndex\n}\n\n/**\n * Returns the last element, or `null` if the
sequence is empty.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.last\n *\npublic fun <T> Sequence<T>.lastOrNull(): T? {\n
val iterator = iterator()\n     if (!iterator.hasNext())\n         return null\n     var last = iterator.next()\n
while (iterator.hasNext())\n         last = iterator.next()\n     return last\n}\n\n/**\n * Returns the last element matching the
given [predicate], or `null` if no such element was found.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.last\n *\npublic inline fun <T> Sequence<T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n     var last: T? = null\n     for (element in this) {\n         if (predicate(element)) {\n             last =
element\n         }\n     }\n     return last\n}\n\n/**\n * Returns the single element, or throws an exception if the
sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n *\npublic fun <T>
Sequence<T>.single(): T {\n     val iterator = iterator()\n     if (!iterator.hasNext())\n         throw
NoSuchElementException("Sequence is empty.")\n     val single = iterator.next()\n     if (iterator.hasNext())\n         throw
IllegalArgumentException("Sequence has more than one element.")\n     return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is
no or more than one matching element.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T>
Sequence<T>.single(predicate: (T) -> Boolean): T {\n     var single: T? = null\n     var found = false\n     for (element
in this) {\n         if (predicate(element)) {\n             if (found) throw IllegalArgumentException("Sequence contains
more than one matching element.")\n             single = element\n             found = true\n         }\n     }\n     if (!found)
throw NoSuchElementException("Sequence contains no element matching the predicate.")\n     @Suppress("UNCHECKED_CAST")\n     return single as T\n}\n\n/**\n * Returns single element, or `null` if the
sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n *\npublic
fun <T> Sequence<T>.singleOrNull(): T? {\n     val iterator = iterator()\n     if (!iterator.hasNext())\n         return
null\n     val single = iterator.next()\n     if (iterator.hasNext())\n         return null\n     return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one
element was found.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T>
Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n     var single: T? = null\n     var found = false\n     for
(element in this) {\n         if (predicate(element)) {\n             if (found) return null\n             single = element\n
found = true\n         }\n     }\n     if (!found) return null\n     return single\n}\n\n/**\n * Returns a sequence containing
all elements except first [n] elements.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun <T> Sequence<T>.drop(n: Int): Sequence<T> {\n     require(n >= 0) { "Requested element count $n
is less than zero." }\n     return when {\n         n == 0 -> this\n         this is DropTakeSequence -> this.drop(n)\n
else -> DropSequence(this, n)\n     }\n}\n\n/**\n * Returns a sequence containing all elements except first elements
that satisfy the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun <T> Sequence<T>.dropWhile(predicate: (T)
-> Boolean): Sequence<T> {\n     return DropWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence

```



```

{\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  return when {\n    n == 0 ->
emptySequence()\n    this is DropTakeSequence -> this.take(n)\n    else -> TakeSequence(this, n)\n  }\n}\n\n/**\n * Returns a sequence containing first elements satisfying the given [predicate].\n *\n * The operation
is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Collections.Transformations.take\n
*/\n\npublic fun <T> Sequence<T>.takeWhile(predicate:
(T) -> Boolean): Sequence<T> {\n  return TakeWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence
that yields elements of this sequence sorted according to their natural sort order.\n *\n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is
_intermediate_ and _stateful_.\n *\n */\n\npublic fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {\n
return object : Sequence<T> {\n  override fun iterator(): Iterator<T> {\n    val sortedList =
this@sorted.toList()\n    sortedList.sort()\n    return sortedList.iterator()\n  }\n}\n}\n\n/**\n
* Returns a sequence that yields elements of this sequence sorted according to natural sort order of the value
returned by specified [selector] function.\n *\n * The sort is _stable_. It means that equal elements preserve their
order relative to each other after sorting.\n *\n * The operation is
_intermediate_ and _stateful_.\n *\n * @sample samples.collections.Collections.Sorting.sortedBy\n
*/\n\npublic inline fun <T, R : Comparable<R>> Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {\n
return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to natural sort order of the value returned by specified [selector] function.\n *\n * The
sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The
operation is _intermediate_ and _stateful_.\n *\n */\n\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.sortedByDescending(crossinline selector: (T) -> R?): Sequence<T> {\n  return
sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to their natural sort order.\n *\n * The sort is _stable_. It means that equal elements
preserve their order relative
to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n *\n */\n\npublic fun <T :
Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {\n  return
sortedWith(reverseOrder())\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according
to the specified [comparator].\n *\n * The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n *\n */\n\npublic fun <T>
Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {\n  return object : Sequence<T> {\n
override fun iterator(): Iterator<T> {\n    val sortedList = this@sortedWith.toList()\n
sortedList.sortWith(comparator)\n    return sortedList.iterator()\n  }\n}\n}\n\n/**\n * Returns a [Map]
containing key-value pairs provided by [transform] function\n *\n * applied to elements of the given sequence.\n *\n *
If any
of two pairs would have the same key the last one gets added to the map.\n *\n * The returned map preserves the
entry iteration order of the original sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.associate\n
*/\n\npublic inline fun <T, K, V>
Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n  return associateTo(LinkedHashMap<K,
V>(), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given sequence indexed by the
key\n *\n * returned from [keySelector] function applied to each element.\n *\n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n *\n * The returned map preserves the entry
iteration order of the original sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.associateBy\n
*/\n\npublic inline fun <T, K>
Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n
return associateByTo(LinkedHashMap<K, T>(), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given
sequence.\n *\n * If any two elements would have the same key returned by [keySelector] the last one gets added to
the map.\n *\n * The returned map preserves the entry iteration order of the original sequence.\n *\n * The

```

operation is `_terminal_`.
`samples.collections.Collections.Transformations.associateByWithValueTransform`
`Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {`
`return`
`associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)}`
`Populates and returns the`
`[destination] mutable map with key-value pairs,`
`where key is provided by the [keySelector] function applied to`
`each element of the given sequence`
`and value is the element itself.`
`If any`
`two elements would have the same key returned by [keySelector] the last one gets added to the map.`
`The`
`operation is _terminal_.`
`@sample`
`samples.collections.Collections.Transformations.associateByTo`
`public inline fun <T, K, M : MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M,`
`keySelector: (T) -> K): M {`
`for (element in this) {`
`destination.put(keySelector(element), element)`
`}`
`return destination}`
`Populates and returns the [destination] mutable map with key-value pairs,`
`where`
`key is provided by the [keySelector] function and`
`and value is provided by the [valueTransform] function`
`applied to elements of the given sequence.`
`If any two elements would have the same key returned by`
`[keySelector] the last one gets added to the map.`
`The operation is _terminal_.`
`@sample`
`samples.collections.Collections.Transformations.associateByToWithValueTransform`
`public inline fun <T, K,`
`V, M : MutableMap<in`
`K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {`
`for (element in this) {`
`destination.put(keySelector(element), valueTransform(element))`
`}`
`return`
`destination}`
`Populates and returns the [destination] mutable map with key-value pairs`
`provided by`
`[transform] function applied to each element of the given sequence.`
`If any of two pairs would have the same`
`key the last one gets added to the map.`
`The operation is _terminal_.`
`@sample`
`samples.collections.Collections.Transformations.associateTo`
`public inline fun <T, K, V, M : MutableMap<in`
`K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {`
`for (element in this)`
`{`
`destination += transform(element)`
`}`
`return destination}`
`Returns a [Map] where keys are`
`elements from the given sequence and values are`
`produced by the [valueSelector] function`
`applied to each element.`
`If any two elements are equal, the last one gets added to the map.`
`The`
`returned map preserves the entry iteration order of the original sequence.`
`The operation is _terminal_.`
`@sample`
`samples.collections.Collections.Transformations.associateWith`
`@SinceKotlin("1.3")`
`public`
`inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {`
`val result =`
`LinkedHashMap<K, V>()`
`return associateWithTo(result, valueSelector)}`
`Populates and returns the`
`[destination] mutable map with key-value pairs for each element of the given sequence,`
`where key is the element`
`itself and value is provided by the [valueSelector] function applied to that key.`
`If any two elements are`
`equal, the last one overwrites the former value in the map.`
`The operation is _terminal_.`
`@sample`
`samples.collections.Collections.Transformations.associateWithTo`
`@SinceKotlin("1.3")`
`public`
`inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K)`
`-> V): M {`
`for (element in this) {`
`destination.put(element, valueSelector(element))`
`}`
`return`
`destination}`
`Appends all elements to the given [destination] collection.`
`The operation is`
`_terminal_.`
`public fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C {`
`for (item in this) {`
`destination.add(item)`
`}`
`return destination}`
`Returns a new [HashSet] of`
`all elements.`
`The operation is _terminal_.`
`public fun <T> Sequence<T>.toHashSet(): HashSet<T> {`
`return toCollection(HashSet<T>())}`
`Returns a [List] containing all elements.`
`The operation is`
`_terminal_.`
`public fun <T> Sequence<T>.toList(): List<T> {`
`return`
`this.toMutableList().optimizeReadOnlyList()}`
`Returns a new [MutableList] filled with all elements of`
`this`
`sequence.`
`The operation is _terminal_.`
`public fun <T> Sequence<T>.toMutableList():`
`MutableList<T> {`
`return toCollection(ArrayList<T>())}`
`Returns a [Set] of all elements.`
`The`
`returned set preserves the element iteration order of the original sequence.`
`The operation is`
`_terminal_.`
`public fun <T> Sequence<T>.toSet(): Set<T> {`
`return`

```

toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()\n\n/**\n * Returns a single sequence of all elements
from results of [transform] function being invoked on each element of original sequence.\n *\n * The operation is
_intermediate_ and _stateless_.\n *\n * @sample samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIterable")\npublic fun <T, R>
Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {\n    return FlatteningSequence(this,
transform, Iterable<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements from results of [transform]
function being invoked on each element of original sequence.\n *\n * The operation is _intermediate_ and
_stateless_.\n *\n * @sample samples.collections.Collections.Transformations.flatMap\n
*\n@kotlin.jvm.JvmName("flatMap")\npublic fun <T, R>
Sequence<T>.flatMap(transform: (T) -> Sequence<R>): Sequence<R> {\n    return FlatteningSequence(this,
transform, Sequence<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original sequence.\n *\n * The operation
is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\npublic fun <T, R>
Sequence<T>.flatMapIndexed(transform:
(index: Int, T) -> Iterable<R>): Sequence<R> {\n    return flatMapIndexed(this, transform,
Iterable<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements yielded from results of [transform]
function being invoked on each element\n * and its index in the original sequence.\n *\n * The operation is
_intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\npublic fun <T, R>
Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): Sequence<R> {\n    return
flatMapIndexed(this, transform, Sequence<R>::iterator)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original sequence, to the given
[destination].\n *\n * The
operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npubl
ic
inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform: (index:
Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIterableTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n    for
(element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return

```



```

destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element of original sequence, to the given [destination].\n *\n
* The operation is _terminal_.\n *\npublic inline fun <T, R, C : MutableCollection<in R>>
Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n  for (element in this) {\n    val
list = transform(element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Groups elements of
the original sequence by the key returned by the given [keySelector] function\n * applied to each element and
returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map
preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T,
K> Sequence<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to
each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The
returned map preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The
operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <T, K, V>
Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements
of the original sequence by the key returned by the given [keySelector] function\n * applied to each element and
puts to the [destination] map each group key associated with a list of corresponding
elements.\n * \n * @return The [destination] map.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T, K, M : MutableMap<in K,
MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n  for (element in this)
{\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) { ArrayList<T>() }\n
list.add(element)\n  }\n  return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original sequence\n * by the key returned by the given [keySelector] function applied
to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n
*\n * @return The [destination] map.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <T, K, V, M :
MutableMap<in
K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V):
M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<V>() }\n    list.add(valueTransform(element))\n  }\n  return destination\n}\n\n/**\n * Creates a
[Grouping] source from a sequence to be used later with one of group-and-fold operations\n * using the specified
[keySelector] function to extract a key from each element.\n *\n * The operation is _intermediate_ and _stateless_.\n
*\n * @sample samples.collections.Grouping.groupingByEachCount\n *\n@SinceKotlin("1.1")\npublic inline
fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n  return object :
Grouping<T, K> {\n    override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n    override fun
keyOf(element: T): K = keySelector(element)\n  }\n}\n\n/**\n * Returns a sequence
containing the results of applying the given [transform] function\n * to each element in the original sequence.\n *\n
* The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.map\n *\npublic fun <T, R> Sequence<T>.map(transform: (T) ->
R): Sequence<R> {\n  return TransformingSequence(this, transform)\n}\n\n/**\n * Returns a sequence containing
the results of applying the given [transform] function\n * to each element and its index in the original sequence.\n
*\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n *\n * The operation is _intermediate_ and _stateless_.\n *\npublic fun <T,
R> Sequence<T>.mapIndexed(transform: (index: Int, T) -> R): Sequence<R> {\n  return
TransformingIndexedSequence(this, transform)\n}\n\n/**\n * Returns a sequence containing only the non-null

```

results of applying the given [transform] function

* to each element and its index in the original sequence.

* @param [transform] function that takes the index of an element and the element itself

* and returns the result of the transform applied to the element.

* The operation is `_intermediate_` and `_stateless_`.

```

public fun <T, R : Any>
Sequence<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): Sequence<R> {
    return TransformingIndexedSequence(this, transform).filterNotNull()
}

```

* Applies the given [transform] function to each element and its index in the original sequence

* and appends only the non-null results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself

* and returns the result of the transform applied to the element.

* The operation is `_terminal_`.

```

public inline fun <T, R : Any, C : MutableCollection<in R>>
Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original sequence

* and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself

* and returns the result of the transform applied to the element.

* The operation is `_terminal_`.

```

public inline fun <T, R, C : MutableCollection<in R>>
Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}

```

* Returns a sequence containing only the non-null results of applying the given [transform] function

* to each element in the original sequence.

* The operation is `_intermediate_` and `_stateless_`.

* @sample

```

samples.collections.Collections.Transformations.mapNotNull

```

```

public fun <T, R : Any>
Sequence<T>.mapNotNull(transform: (T) -> R?): Sequence<R> {
    return TransformingSequence(this, transform).filterNotNull()
}

```

* Applies the given [transform] function to each element in the original sequence

* and appends only the non-null results to the given [destination].

* The operation is `_terminal_`.

```

public inline fun <T, R : Any, C : MutableCollection<in R>>
Sequence<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) } }
    return destination
}

```

* Applies the given [transform] function to each element of the original sequence

* and appends the results to the given [destination].

* The operation is `_terminal_`.

```

public inline fun <T, R, C : MutableCollection<in R>>
Sequence<T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

```

* Returns a sequence that wraps each element of the original sequence

* into an [IndexedValue] containing the index of that element and the element itself.

* The operation is `_intermediate_` and `_stateless_`.

```

public fun <T>
Sequence<T>.withIndex(): Sequence<IndexedValue<T>> {
    return IndexingSequence(this)
}

```

* Returns a sequence containing only distinct elements from the given sequence.

* Among equal elements of the given sequence, only the first one will be present in the resulting sequence.

* The elements in the resulting sequence are in the same order as they were in the source sequence.

* The operation is `_intermediate_` and `_stateful_`.

* @sample

```

samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun <T>
Sequence<T>.distinct(): Sequence<T> {
    return this.distinctBy { it }
}

```

* Returns a sequence containing only elements from the given sequence

* having distinct keys returned by the given [selector] function.

* Among elements of the given sequence with equal keys, only the first one will be present in the resulting sequence.

* The elements in the resulting sequence are in the same order as they were in the source sequence.

* The operation is `_intermediate_` and `_stateful_`.

* @sample

```

samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun <T, K>
Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {
    return DistinctSequence(this, selector)
}

```

* Returns a new [MutableSet] containing all distinct elements from the given sequence.

* The returned set preserves the element iteration order of the original sequence.

* The operation is `_terminal_`.

```

<T> Sequence<T>.toMutableSet(): MutableSet<T> {\n    val set = LinkedHashSet<T>()\n    for (item in this)
set.add(item)\n    return set}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n
*\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic
inline fun <T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean {\n    for (element in this) if
(!predicate(element)) return false\n    return true}\n\n/**\n * Returns `true` if sequence has at least one element.\n
*\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic
fun <T> Sequence<T>.any(): Boolean {\n    return iterator().hasNext()}\n\n/**\n * Returns `true` if at least one
element matches the given [predicate].\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <T>
Sequence<T>.any(predicate: (T) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
true\n    return false}\n\n/**\n * Returns the number of elements in this sequence.\n * \n * The operation is
_terminal_.\n */\npublic fun
<T> Sequence<T>.count(): Int {\n    var count = 0\n    for (element in this) checkCountOverflow(++count)\n
return count}\n\n/**\n * Returns the number of elements matching the given [predicate].\n * \n * The operation is
_terminal_.\n */\npublic inline fun <T> Sequence<T>.count(predicate: (T) -> Boolean): Int {\n    var count = 0\n
for (element in this) if (predicate(element)) checkCountOverflow(++count)\n    return count}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator
value and each element.\n * \n * Returns the specified [initial] value if the sequence is empty.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * The operation is _terminal_.\n */\npublic inline fun <T, R> Sequence<T>.fold(initial: R, operation:
(acc: R, T) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator,
element)\n    return accumulator}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from left to right\n * to current accumulator value and each element with its index in the original
sequence.\n * \n * Returns the specified [initial] value if the sequence is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * The operation is _terminal_.\n */\npublic inline fun <T, R>
Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n    var index = 0\n    var
accumulator = initial\n    for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator,
element)\n    return accumulator}\n\n/**\n * Performs the given [action] on each element.\n * \n * The operation is
_terminal_.\n */\npublic inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {\n    for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential
index with the element.\n * \n * @param [action] function that takes the index of an element and the element itself\n
* and performs the action on the element.\n * \n * The operation is _terminal_.\n */\npublic inline fun <T>
Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(checkIndexOverflow(index++), item)\n}\n\n@Deprecated("Use maxOrNull instead.")
ReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Sequence<Double>.max(): Double? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.")
ReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Sequence<Float>.max(): Float? {\n
return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.")
ReplaceWith("this.maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Sequence<T>.max(): T? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.")
ReplaceWith("this.maxByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxBy(selector: (T) ->
R): T? {\n    return maxByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the largest value of the
given function or `null` if there are no elements.\n * \n * The operation is _terminal_.\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.maxByOrNull\n *\\n@SinceKotlin(\\\"1.4\\\")\\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.maxByOrNull(selector: (T) -> R): T? {\\n  val iterator = iterator()\\n  if
(!iterator.hasNext()) return
null\\n  var maxElem = iterator.next()\\n  if (!iterator.hasNext()) return maxElem\\n  var maxV alue =
selector(maxElem)\\n  do {\\n    val e = iterator.next()\\n    val v = selector(e)\\n    if (maxV alue < v) {\\n
maxElem = e\\n    maxV alue = v\\n    }\\n  } while (iterator.hasNext())\\n  return maxElem\\n}\\n\\n/**\\n *
Returns the largest value among all values produced by [selector] function\\n * applied to each element in the
sequence.\\n * \\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\\n * \\n *
@throws NoSuchElementException if the sequence is empty.\\n * \\n * The operation is _terminal_.\\n
*\\n@SinceKotlin(\\\"1.4\\\")\\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\\n@OverloadResolution
ByLambdaReturnType\\n@kotlin.internal.InlineOnly\\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) ->
Double): Double {\\n  val iterator = iterator()\\n  if (!iterator.hasNext()) throw NoSuchElementException()\\n
  var maxV alue = selector(iterator.next())\\n  while (iterator.hasNext()) {\\n    val v = selector(iterator.next())\\n
maxV alue = maxOf(maxV alue, v)\\n  }\\n  return maxV alue\\n}\\n\\n/**\\n * Returns the largest value among all
values produced by [selector] function\\n * applied to each element in the sequence.\\n * \\n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\\n * \\n * @throws NoSuchElementException
if the sequence is empty.\\n * \\n * The operation is _terminal_.\\n
*\\n@SinceKotlin(\\\"1.4\\\")\\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\\n@OverloadResolution
ByLambdaReturnType\\n@kotlin.internal.InlineOnly\\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) ->
Float): Float {\\n  val iterator = iterator()\\n  if (!iterator.hasNext()) throw NoSuchElementException()\\n  var
maxV alue = selector(iterator.next())\\n  while (iterator.hasNext()) {\\n    val v = selector(iterator.next())\\n
maxV alue = maxOf(maxV alue,
v)\\n  }\\n  return maxV alue\\n}\\n\\n/**\\n * Returns the largest value among all values produced by [selector]
function\\n * applied to each element in the sequence.\\n * \\n * @throws NoSuchElementException if the sequence is
empty.\\n * \\n * The operation is _terminal_.\\n
*\\n@SinceKotlin(\\\"1.4\\\")\\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\\n@OverloadResolution
ByLambdaReturnType\\n@kotlin.internal.InlineOnly\\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOf(selector: (T) -> R): R {\\n  val iterator = iterator()\\n  if (!iterator.hasNext()) throw
NoSuchElementException()\\n  var maxV alue = selector(iterator.next())\\n  while (iterator.hasNext()) {\\n    val v
= selector(iterator.next())\\n    if (maxV alue < v) {\\n      maxV alue = v\\n    }\\n  }\\n  return
maxV alue\\n}\\n\\n/**\\n * Returns the largest value among all values produced by [selector] function\\n * applied to
each element in the sequence or `null` if there are no elements.\\n
* \\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\\n * \\n * The operation
is _terminal_.\\n
*\\n@SinceKotlin(\\\"1.4\\\")\\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\\n@OverloadResolution
ByLambdaReturnType\\n@kotlin.internal.InlineOnly\\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector:
(T) -> Double): Double? {\\n  val iterator = iterator()\\n  if (!iterator.hasNext()) return null\\n  var maxV alue =
selector(iterator.next())\\n  while (iterator.hasNext()) {\\n    val v = selector(iterator.next())\\n    maxV alue =
maxOf(maxV alue, v)\\n  }\\n  return maxV alue\\n}\\n\\n/**\\n * Returns the largest value among all values produced
by [selector] function\\n * applied to each element in the sequence or `null` if there are no elements.\\n * \\n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\\n * \\n * The operation is _terminal_.\\n
*\\n@SinceKotlin(\\\"1.4\\\")\\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\\n@OverloadResolution
ByLambdaReturnType\\n@kotlin.internal.InlineOnly\\npublic
inline fun <T> Sequence<T>.maxOfOrNull(selector: (T) -> Float): Float? {\\n  val iterator = iterator()\\n  if
(!iterator.hasNext()) return null\\n  var maxV alue = selector(iterator.next())\\n  while (iterator.hasNext()) {\\n
val v = selector(iterator.next())\\n    maxV alue = maxOf(maxV alue, v)\\n  }\\n  return maxV alue\\n}\\n\\n/**\\n *
Returns the largest value among all values produced by [selector] function\\n * applied to each element in the
sequence or `null` if there are no elements.\\n * \\n * The operation is _terminal_.\\n

```



```

minOrNull()\n\n@Deprecated(\n"Use minOrNull instead.\n",
ReplaceWith(\n"this.minOrNull()\n")\n@DeprecatedSinceKotlin(warningSince = \n"1.4\n", errorSince = \n"1.5\n",
hiddenSince = \n"1.6\n")\npublic fun <T : Comparable<T>> Sequence<T>.min(): T? {\n    return
minOrNull()\n}\n\n@Deprecated(\n"Use minByOrNull instead.\n",
ReplaceWith(\n"this.minByOrNull(selector)\n")\n@DeprecatedSinceKotlin(warningSince
= \n"1.4\n", errorSince = \n"1.5\n", hiddenSince = \n"1.6\n")\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minBy(selector: (T) -> R): T? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * The operation is
_terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n@SinceKotlin(\n"1.4\n")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minByOrNull(selector: (T) ->
R): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minElem = iterator.next()\n    if
(!iterator.hasNext()) return minElem\n    var minValue = selector(minElem)\n    do {\n        val e = iterator.next()\n
        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    } while
(iterator.hasNext())\n    return minElem\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the
sequence is empty.\n * \n * The operation is _terminal_.\n
*/\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values
produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * The
operation is _terminal_.\n
*/\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * The operation is _terminal_.\n
*/\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Comparable<R>> Sequence<T>.minOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n
        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is _terminal_.\n
*/\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOfOrNull(selector:
(T) -> Double): Double? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector]

```

```

function is `NaN`, the returned result is `NaN`.
 * Since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOrNull(selector:
 (T) -> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
 selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue
 = minOf(minValue, v)\n    }\n    return minValue\n}\n\n * Returns the smallest value among all values
 produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.
 The operation is _terminal_.
 * Since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
 Sequence<T>.minOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
 null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
 selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return
 minValue\n}\n\n * Returns the smallest value according to the provided [comparator]\n * among all values
 produced by [selector] function applied to each element in the sequence.\n * @throws
 NoSuchElementException
 if the sequence is empty.
 The operation is _terminal_.
 * Since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <T, R>
 Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if
 (!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while
 (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n * Returns the smallest value according to the
 provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence
 or `null` if there are no elements.
 The operation is _terminal_.
 * Since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType@kotlin.internal.InlineOnly\npublic
 inline fun <T, R> Sequence<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n
 val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n
 while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n * Returns the smallest element or `null` if there are
 no elements.\n * If any of elements is `NaN` returns `NaN`.
 The operation is _terminal_.
 * Since Kotlin("1.4")\npublic fun Sequence<Double>.minOrNull(): Double? {\n    val iterator = iterator()\n    if
 (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
 iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n * Returns the smallest
 element or `null` if there are no elements.\n * If any of elements is `NaN` returns `NaN`.
 The operation
 is _terminal_.
 * Since Kotlin("1.4")\npublic fun Sequence<Float>.minOrNull(): Float? {\n    val iterator =
 iterator()\n    if (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n
 val e = iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n * Returns the smallest element or
 `null` if there are no elements.
 The operation is _terminal_.
 * Since Kotlin("1.4")\npublic fun <T :
 Comparable<T>> Sequence<T>.minOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
 null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (min > e) min
 = e\n    }\n    return min\n}\n\n@Deprecated("Use minWithOrNull instead.")
 ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince
 = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun <T> Sequence<T>.minWith(comparator:
 Comparator<in T>): T? {\n    return minWithOrNull(comparator)\n}\n\n * Returns the first element having the
 smallest value according to the provided [comparator] or `null` if there are no elements.
 The operation is
 _terminal_.
 * Since Kotlin("1.4")\npublic fun <T> Sequence<T>.minWithOrNull(comparator:

```

```

Comparator<in T>: T? {\n  val iterator = iterator()\n  if (!iterator.hasNext()) return null\n  var min =
iterator.next()\n  while (iterator.hasNext()) {\n    val e = iterator.next()\n    if (comparator.compare(min, e) >
0) min = e\n  }\n  return min\n}\n\n * Returns `true` if the sequence has no elements.\n * \n * The operation is
_terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun <T>
Sequence<T>.none(): Boolean {\n  return !iterator().hasNext()\n}\n\n **\n * Returns `true` if no elements match
the
given [predicate].\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <T>
Sequence<T>.none(predicate: (T) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return
false\n  return true\n}\n\n **\n * Returns a sequence which performs the given [action] on each element of the
original sequence as they pass through it.\n * \n * The operation is _intermediate_ and _stateless_.\n
*\n */\n@SinceKotlin("1.1")\npublic fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {\n  return
map {\n    action(it)\n    it\n  }\n}\n\n **\n * Returns a sequence which performs the given [action] on each
element of the original sequence as they pass through it.\n * \n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n * \n * The operation is _intermediate_
and _stateless_.\n */\n@SinceKotlin("1.4")\npublic
fun <T> Sequence<T>.onEachIndexed(action: (index: Int, T) -> Unit): Sequence<T> {\n  return mapIndexed {
index, element ->\n    action(index, element)\n    element\n  }\n}\n\n **\n * Accumulates value starting with
the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n *
\n * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,\n * please use
[reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * The operation is
_terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S>
Sequence<T>.reduce(operation: (acc: S, T) -> S): S {\n  val iterator = this.iterator()\n  if (!iterator.hasNext())
throw UnsupportedOperationException("Empty sequence can't be reduced.")\n
  var accumulator: S = iterator.next()\n  while (iterator.hasNext()) {\n    accumulator = operation(accumulator,
iterator.next())\n  }\n  return accumulator\n}\n\n **\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original sequence.\n * \n * Throws an exception if this sequence is empty. If the sequence can be empty in an
expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value and the element itself,\n
* and calculates the next accumulator value.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S>
Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n  val iterator = this.iterator()\n  if
(!iterator.hasNext())\n  throw UnsupportedOperationException("Empty sequence can't be reduced.")\n  var index = 1\n  var
accumulator: S = iterator.next()\n  while (iterator.hasNext()) {\n    accumulator =
operation(checkIndexOverflow(index++), accumulator, iterator.next())\n  }\n  return accumulator\n}\n\n **\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element with its index in the original sequence.\n * \n * Returns `null` if the sequence is
empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the
element itself,\n * and calculates the next accumulator value.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <S, T : S>
Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n  val iterator = this.iterator()\n
  if (!iterator.hasNext()) return null\n  var index = 1\n  var accumulator: S = iterator.next()\n  while
(iterator.hasNext()) {\n    accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n
  }\n  return accumulator\n}\n\n **\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the sequence is

```


empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n    val iterator = this.iterator()\n    if
(!iterator.hasNext()) return null\n    var accumulator: S = iterator.next()\n    while (iterator.hasNext()) {\n        accumulator = operation(accumulator, iterator.next())\n    }\n    return
accumulator\n}

```

\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```

*\n@SinceKotlin("1.4")\npublic fun <T, R>
Sequence<T>.runningFold(initial:
R, operation: (acc: R, T) -> R): Sequence<R> {\n    return sequence {\n        yield(initial)\n        var accumulator =
initial\n        for (element in this@runningFold) {\n            accumulator = operation(accumulator, element)\n            yield(accumulator)\n        }\n    }\n}

```

\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```

*\n@SinceKotlin("1.4")\npublic fun <T, R>
Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n    return
sequence {\n        yield(initial)\n        var index = 0\n        var accumulator = initial\n        for (element in
this@runningFoldIndexed) {\n            accumulator = operation(checkIndexOverflow(index++), accumulator,
element)\n            yield(accumulator)\n        }\n    }\n}

```

\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * \n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <S, T : S>
Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {\n    return sequence {\n        val iterator =
iterator()\n        if (iterator.hasNext()) {\n            var accumulator: S = iterator.next()\n            yield(accumulator)\n            while (iterator.hasNext()) {\n                accumulator = operation(accumulator, iterator.next())\n            }\n            yield(accumulator)\n        }\n    }\n}

```

\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the


```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence<T>.sumOf(selector: (T) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing
an [IllegalArgumentException] if there are any `null` elements.\n * \n * The operation is _intermediate_ and
_stateless_.\n */\npublic fun <T : Any> Sequence<T?>.requireNonNulls(): Sequence<T> {\n    return map { it ?:
throw IllegalArgumentException("null element found in $this.") }\n}\n\n/**\n * Splits this sequence into a
sequence of lists each not exceeding the given [size].\n * \n * The last list in the resulting sequence may have fewer
elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive
and can be greater than the number of elements in this sequence.\n * \n * The operation is _intermediate_ and
_stateful_.\n * \n * @sample samples.collections.Collections.Transformations.chunked\n
*\n@SinceKotlin("1.2")\npublic fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {\n    return
windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this sequence into several lists each not exceeding
the given [size]\n *
and applies the given [transform] function to an each.\n * \n * @return sequence of results of the [transform]
applied to an each list.\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only
inside that function.\n * \n * You should not store it or allow it to escape in some way, unless you made a snapshot of
it.\n * \n * The last list may have fewer elements than the given [size].\n * \n * @param size the number of elements to
take in each list, must be positive and can be greater than the number of elements in this sequence.\n * \n * The
operation is _intermediate_ and _stateful_.\n * \n * @sample samples.text.Strings.chunkedTransform\n
*\n@SinceKotlin("1.2")\npublic fun <T, R> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R):
Sequence<R> {\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n *
Returns a sequence containing all elements of the original sequence without the first occurrence of the given
[element].\n
*\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.minus(element:
T): Sequence<T> {\n    return object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n            var
removed = false\n            return this@minus.filter { if (!removed && it == element) { removed = true; false } else
true }.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except
the elements contained in the given [elements] array.\n * \n * Note that the source sequence and the array being
subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them
between successive calls to `iterator` may affect the result.\n * \n * Before Kotlin 1.6, the [elements] array may have
been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and
stable implementation of `hashCode()` that didn't change between
successive invocations.\n * \n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n * \n * The operation is _intermediate_ and
_stateful_.\n */\npublic operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {\n    if

```

```

(elements.isEmpty()) return this\n    return object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n            val other = elements.convertToSetForSetOperation()\n            return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements\n contained in the given [elements] collection.\n * Note that the source sequence and the collection being\n subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them\n between successive calls to `iterator` may affect the result.\n * Before Kotlin 1.6, the\n [elements] collection may have been converted to a [HashSet] to speed up the operation, thus the elements were\n required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive\n invocations.\n * On JVM, you can enable this behavior back with the system property\n `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\n * The operation is _intermediate_ and\n _stateful_.  

\n\npublic operator fun <T> Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {\n    return\n object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n            val other =\n elements.convertToSetForSetOperation()\n            if (other.isEmpty())\n                return this@minus.iterator()\n            else\n                return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence\n containing all elements of original sequence except the elements contained in the given [elements] sequence.\n * Note that the source sequence and the sequence being subtracted are iterated only when an `iterator` is requested\n from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n * The operation is _intermediate_ for this sequence and _terminal_ and _stateful_ for the [elements] sequence.\n * Before Kotlin 1.6, the [elements] sequence may have been converted to a [HashSet] to speed up the operation,\n thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change\n between successive invocations.\n * On JVM, you can enable this behavior back with the system property\n `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\n\npublic operator fun <T>\n Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {\n    return object: Sequence<T> {\n        override fun\n iterator(): Iterator<T> {\n            val other = elements.convertToSetForSetOperation()\n            if (other.isEmpty())\n                return this@minus.iterator()\n            else\n                return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence containing all elements of the original\n sequence without the first occurrence of the given [element].\n * The operation is _intermediate_ and\n _stateless_.  

\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minusElement(element: T):\n Sequence<T> {\n    return minus(element)\n}\n\n/**\n * Splits the original sequence into pair of lists,\n * where\n *first* list contains elements for which [predicate] yielded `true`,\n * while\n *second* list contains elements for\n which [predicate] yielded `false`.\n * The operation is _terminal_.  

\n * @sample\n samples.collections.Sequences.Transformations.partition  

\n\npublic inline fun <T>\n Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val\n second = ArrayList<T>()\n\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Returns a sequence containing all\n elements of the original sequence and then the given [element].\n * The operation is _intermediate_ and\n _stateless_.  

\n\npublic operator fun <T> Sequence<T>.plus(element: T): Sequence<T> {\n    return\n sequenceOf(this, sequenceOf(element)).flatten()\n}\n\n/**\n * Returns a sequence containing all elements of\n original sequence and then all elements of the given [elements] array.\n * Note that the source sequence and the\n array being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of\n them between successive calls to `iterator` may affect the result.\n * The operation is _intermediate_ and\n _stateless_.  

\n\npublic operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T>\n {\n    return this.plus(elements.asList())\n}\n\n/**\n * Returns a sequence containing all elements of original\n sequence and then all elements of the given [elements] collection.\n * Note that the source sequence and the\n collection being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing\n any of them between successive calls to `iterator` may affect the result.\n * The operation is _intermediate_ and\n _stateless_.  

\n\npublic operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {\n    return

```

`sequenceOf(this, elements.asSequence()).flatten()`\n\n**\n * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] sequence.\n * \n * Note that the source sequence and the sequence being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n

* The operation is `_intermediate_` and `_stateless_`.\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {\n return sequenceOf(this, elements).flatten()\n}\n\n**\n * Returns a sequence containing all elements of the original sequence and then the given [element].\n *\n * The operation is `_intermediate_` and `_stateless_`.\n *\n

`Sequence<T>.plusElement(element: T): Sequence<T>` {\n return plus(element)\n}\n\n**\n * Returns a sequence of snapshots of the window of the given [size]\n * sliding along this sequence with the given [step], where each\n * snapshot is a list.\n * \n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default

1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample

`samples.collections.Sequences.Transformations.takeWindows`\n *\n@SinceKotlin("1.2")\npublic fun <T>
Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<List<T>> {\n return windowedSequence(size, step, partialWindows, reuseBuffer = false)\n}\n\n**\n * Returns a sequence of results of applying the given [transform] function to\n * an each list representing a view over the window of the given [size]\n * sliding along this sequence with the given [step].\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step]

must be positive and can be greater than the number of elements in this sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample

`samples.collections.Sequences.Transformations.averageWindows`\n *\n@SinceKotlin("1.2")\npublic fun <T, R>
Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R):

Sequence<R> {\n return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)\n}\n\n**\n * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n *\n * The operation is `_intermediate_` and

`_stateless_`.\n *\n * @sample `samples.collections.Sequences.Transformations.zip`\n *\npublic infix fun <T, R>
Sequence<T>.zip(other: Sequence<R>): Sequence<Pair<T, R>> {\n return MergingSequence(this, other) { t1, t2 -> t1 to t2 }\n}\n\n**\n * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n *\n * The operation is `_intermediate_` and

`_stateless_`.\n *\n * @sample `samples.collections.Sequences.Transformations.zipWithTransform`\n *\npublic fun

<T, R, V> Sequence<T>.zip(other: Sequence<R>, transform: (a: T, b: R) -> V): Sequence<V> {\n return MergingSequence(this, other, transform)\n}\n\n**\n * Returns a sequence of pairs of each two adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two elements.\n *\n * The operation is `_intermediate_` and `_stateless_`.\n *\n * @sample

`samples.collections.Collections.Transformations.zipWithNext`\n *\n@SinceKotlin("1.2")\npublic fun <T>
Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {\n return zipWithNext { a, b -> a to b }\n}\n\n**\n * Returns a sequence containing the results of applying the given [transform] function\n * to an each pair of two adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two

```

elements.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n *\n@SinceKotlin("1.2")\npublic
fun <T, R> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R): Sequence<R> {\n  return sequence result@
{\n    val iterator = iterator()\n    if (!iterator.hasNext()) return @result\n    var current = iterator.next()\n
while (iterator.hasNext()) {\n    val next = iterator.next()\n
    yield(transform(current, next))\n    current = next\n    }\n }\n}\n\n/**\n * Appends the string from
all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n *\n * If the
collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n *
elements will be appended, followed by the [truncated] string (which defaults to "...").\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Transformations.joinTo\n *\n\npublic fun <T, A :
Appendable> Sequence<T>.joinTo(buffer: A, separator: CharSequence = '\n', prefix: CharSequence = "", postfix:
CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A
{\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit)
{\n      buffer.appendElement(element, transform)\n    } else break\n  }\n  if (limit >= 0 && count > limit)
buffer.append(truncated)\n  buffer.append(postfix)\n  return buffer\n}\n\n/**\n * Creates a string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n *\n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will
be appended, followed by the [truncated] string (which defaults to "...").\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Transformations.joinToString\n *\n\npublic fun <T>
Sequence<T>.joinToString(separator: CharSequence = '\n', prefix: CharSequence = "", postfix: CharSequence =
"\n", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {\n  return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n *
Creates an [Iterable] instance that wraps the original sequence returning its elements when being iterated.\n
*\n\npublic fun <T> Sequence<T>.asIterable(): Iterable<T> {\n  return Iterable { this.iterator() }\n}\n\n/**\n *
Returns this sequence as a [Sequence].\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.asSequence(): Sequence<T> {\n  return this\n}\n\n/**\n * Returns an average value of elements in
the sequence.\n *\n * The operation is _terminal_.\n *\n\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun
Sequence<Byte>.average(): Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n
sum += element\n    checkCountOverflow(++count)\n  }\n  return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Sequence<Short>.average():
Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n
checkCountOverflow(++count)\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Sequence<Int>.average(): Double {\n  var sum: Double
= 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    checkCountOverflow(++count)\n
}\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in
the sequence.\n *\n * The operation is _terminal_.\n *\n\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun
Sequence<Long>.average(): Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n
sum += element\n    checkCountOverflow(++count)\n  }\n  return if (count == 0) Double.NaN else
sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is
_terminal_.\n *\n\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Sequence<Float>.average(): Double {\n
var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n
checkCountOverflow(++count)\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Sequence<Double>.average(): Double {\n  var sum:
Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n

```

```

checkCountOverflow(++count)\n } \n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Sequence<Byte>.sum(): Int
{\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns
the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Sequence<Short>.sum(): Int {\n var sum: Int = 0\n for
(element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the
sequence.\n *\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun
Sequence<Int>.sum(): Int {\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n return
sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Sequence<Long>.sum(): Long {\n var sum: Long = 0L\n
for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in
the sequence.\n
*\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun
Sequence<Float>.sum(): Float {\n var sum: Float = 0.0f\n for (element in this) {\n sum += element\n }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Sequence<Double>.sum(): Double {\n var sum:
Double = 0.0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n"/\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns a set containing
all elements of the original set except the given [element].\n *\n * The returned set preserves the element iteration
order of the original set.\n *\npublic operator fun <T> Set<T>.minus(element: T): Set<T> {\n val result =
LinkedHashSet<T>(mapCapacity(size))\n var removed = false\n return this.filterTo(result) { if (!removed &&
it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a set containing all elements of the original
set except the elements contained in the given [elements] array.\n *\n * The returned set preserves the element
iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] array may have been converted to a
[HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation
of `hashCode()` that didn't change between successive invocations.\n
*\n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\npublic operator fun <T>
Set<T>.minus(elements: Array<out T>): Set<T> {\n val result = LinkedHashSet<T>(this)\n
result.removeAll(elements)\n return result\n}\n\n/**\n * Returns a set containing all elements of the original set
except the elements contained in the given [elements] collection.\n *\n * The returned set preserves the element
iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] collection may have been converted to a
[HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation
of `hashCode()` that didn't change between successive invocations.\n *\n * On JVM, you can enable this behavior back
with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\npublic operator
fun <T> Set<T>.minus(elements: Iterable<T>):
Set<T> {\n val other = elements.convertToSetForSetOperationWith(this)\n if (other.isEmpty())\n return
this.toSet()\n if (other is Set)\n return this.filterNotTo(LinkedHashSet<T>()) { it in other }\n val result =
LinkedHashSet<T>(this)\n result.removeAll(other)\n return result\n}\n\n/**\n * Returns a set containing all
elements of the original set except the elements contained in the given [elements] sequence.\n *\n * The returned set
preserves the element iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] sequence may
have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct

```

and stable implementation of `hashCode()` that didn't change between successive invocations.

On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.

```

public operator fun <T> Set<T>.minus(elements: Sequence<T>):
Set<T> {
    val result = LinkedHashSet<T>(this)
    result.removeAll(elements)
    return result
}

```

Returns a set containing all elements of the original set except the given [element].

The returned set preserves the element iteration order of the original set.

```

@kotlin.internal.InlineOnly
public inline fun <T>
Set<T>.minusElement(element: T): Set<T> {
    return minus(element)
}

```

Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.

The returned set preserves the element iteration order of the original set.

```

public operator fun <T> Set<T>.plus(element: T):
Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(size + 1))
    result.addAll(this)
    result.add(element)
    return result
}

```

Returns a set containing all elements of the original set and the given [elements] array, which aren't already in this set.

The returned set preserves the element iteration order of the original set.

```

public operator fun <T> Set<T>.plus(elements: Array<out T>):
Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))
    result.addAll(this)
    result.addAll(elements)
    return result
}

```

Returns a set containing all elements of the original set and the given [elements] collection, which aren't already in this set.

The returned set preserves the element iteration order of the original set.

```

public operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))
    result.addAll(this)
    result.addAll(elements)
    return result
}

```

Returns a set containing all elements of the original set and the given [elements] sequence, which aren't already in this set.

The returned set preserves the element iteration order of the original set.

```

public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {
    val result =
    LinkedHashSet<T>(mapCapacity(this.size * 2))
    result.addAll(this)
    result.addAll(elements)
    return
    result
}

```

Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.

The returned set preserves the element iteration order of the original set.

```

@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.plusElement(element: T): Set<T> {
    return
    plus(element)
}

```

Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

<https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib>

```

kotlin.random.*

```

Returns a character at the given [index] or throws an `IndexOutOfBoundsException` if the [index] is out of bounds of this char sequence.

@sample

```

samples.collections.Collections.Elements.elementAt

```

public expect fun CharSequence.elementAt(index: Int): Char

Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.

@sample

```

samples.collections.Collections.Elements.elementAtOrElse

```

```

@kotlin.internal.InlineOnly
public inline fun
CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

```

Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.

@sample

```

samples.collections.Collections.Elements.elementAtOrNull

```

```

@kotlin.internal.InlineOnly
public
inline fun CharSequence.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)
}

```

Returns the first character matching the given [predicate], or `null` if no such character was found.

@sample

```

samples.collections.Collections.Elements.find

```

```

@kotlin.internal.InlineOnly
public inline fun
CharSequence.find(predicate: (Char) -> Boolean): Char? {
    return firstOrNull(predicate)
}

```

Returns the last character matching the given [predicate], or `null` if no such character was found.

@sample

```

samples.collections.Collections.Elements.find

```

```

@kotlin.internal.InlineOnly
public inline fun

```



```

CharSequence.findLast(predicate: (Char) -> Boolean): Char? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns first character.\n * @throws [NoSuchElementException] if the char sequence is empty.\n */\npublic fun\nCharSequence.first(): Char {\n  if (isEmpty())\n    throw NoSuchElementException("Char sequence is\n  empty.")\n  return this[0]\n}\n\n/**\n * Returns the first character matching the given [predicate].\n * @throws\n [NoSuchElementException] if no such character is found.\n */\npublic inline fun CharSequence.first(predicate:\n (Char) -> Boolean): Char {\n  for (element in this) if (predicate(element)) return element\n  throw\n NoSuchElementException("Char sequence contains no character matching the predicate.")\n}\n\n/**\n * Returns\n the first non-null value produced by [transform] function being applied to characters of this char sequence in\n iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n */\n * @sample\n samples.collections.Collections.Transformations.firstNotNullOf\n\n*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <R : Any>\nCharSequence.firstNotNullOf(transform: (Char) -> R?): R {\n  return firstNotNullOfOrNull(transform) ?: throw\n NoSuchElementException("No element of the char sequence was transformed to a non-null\n  value.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to characters\n of this char sequence in iteration order,\n * or `null` if no non-null value was produced.\n */\n * @sample\n samples.collections.Collections.Transformations.firstNotNullOf\n\n*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <R : Any>\nCharSequence.firstNotNullOfOrNull(transform: (Char) -> R?): R? {\n  for (element in this) {\n    val result =\n  transform(element)\n    if (result != null) {\n      return result\n    }\n  }\n  return null\n}\n\n/**\n * Returns the first character, or `null` if the char sequence is empty.\n */\npublic fun CharSequence.firstOrNull():\n Char? {\n  return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first character matching the given\n [predicate], or `null` if character was not found.\n */\npublic inline fun CharSequence.firstOrNull(predicate: (Char) -\n > Boolean): Char? {\n  for\n (element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns a character at the given\n [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char):\n Char {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns a\n character at the given [index] or `null` if the [index] is out of bounds of this char sequence.\n */\n * @sample\n samples.collections.Collections.Elements.getOrNull\n */\npublic fun CharSequence.getOrNull(index: Int): Char?\n {\n  return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns index of the first\n character matching the given [predicate], or -1 if the char sequence does not contain such character.\n */\npublic\n inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {\n\n  for (index in indices) {\n    if (predicate(this[index])) {\n      return index\n    }\n  }\n  return -\n  1\n}\n\n/**\n * Returns index of the last character matching the given [predicate], or -1 if the char sequence does\n not contain such character.\n */\npublic inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {\n\n  for (index in indices.reversed()) {\n    if (predicate(this[index])) {\n      return index\n    }\n  }\n  return -\n  1\n}\n\n/**\n * Returns the last character.\n */\n * @throws NoSuchElementException if the char sequence is\n empty.\n */\n * @sample\n samples.text.Strings.last\n */\npublic fun CharSequence.last(): Char {\n  if (isEmpty())\n    throw NoSuchElementException("Char sequence is empty.")\n  return this[lastIndex]\n}\n\n/**\n * Returns the\n last character matching the given [predicate].\n */\n * @throws NoSuchElementException if no such character is\n found.\n */\n * @sample\n samples.text.Strings.last\n */\npublic\n inline fun CharSequence.last(predicate: (Char) -> Boolean): Char {\n  for (index in this.indices.reversed()) {\n\n    val element = this[index]\n    if (predicate(element)) return element\n  }\n  throw\n NoSuchElementException("Char sequence contains no character matching the predicate.")\n}\n\n/**\n * Returns\n the last character, or `null` if the char sequence is empty.\n */\n * @sample\n samples.text.Strings.last\n */\npublic fun\nCharSequence.lastOrNull(): Char? {\n  return if (isEmpty()) null else this[length - 1]\n}\n\n/**\n * Returns the last\n character matching the given [predicate], or `null` if no such character was found.\n */\n * @sample\n samples.text.Strings.last\n */\npublic inline fun CharSequence.lastOrNull(predicate: (Char) -> Boolean): Char? {\n

```

```

for (index in this.indices.reversed()) {
    val element = this[index]
    if (predicate(element)) return element
}
return null
}

/**
 * Returns a random character from this char sequence.
 *
 * @throws NoSuchElementException if this char sequence is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun CharSequence.random(): Char {
    return random(Random)
}

/**
 * Returns a random character from this char sequence using the specified
 * source of randomness.
 *
 * @throws NoSuchElementException if this char sequence is empty.
 */
@SinceKotlin("1.3")
public fun CharSequence.random(random: Random): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return get(random.nextInt(length))
}

/**
 * Returns a random character from this char sequence, or `null` if this char sequence is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun CharSequence.randomOrNull(): Char? {
    return randomOrNull(Random)
}

/**
 * Returns a
 * random character from this char sequence using the specified source of
 * randomness, or `null` if this char sequence is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun
CharSequence.randomOrNull(random: Random): Char? {
    if (isEmpty())
        return null
    return
    get(random.nextInt(length))
}

/**
 * Returns the single character, or throws an exception if the char sequence
 * is empty or has more than one character.
 */
public fun CharSequence.single(): Char {
    return when (length)
    {
        0 -> throw NoSuchElementException("Char sequence is empty.")
        1 -> this[0]
        else -> throw
        IllegalArgumentException("Char sequence has more than one element.")
    }
}

/**
 * Returns the single
 * character matching the given [predicate], or throws exception if there is no or more than one matching character.
 */
public inline fun CharSequence.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element))
            if (found) throw IllegalArgumentException("Char sequence contains more than one matching
            element.")
            single = element
            found = true
    }
    if (!found) throw
    NoSuchElementException("Char sequence contains no character matching the predicate.")
}

@Suppress("UNCHECKED_CAST")
return single as Char

/**
 * Returns single character, or `null` if
 * the char sequence is empty or has more than one character.
 */
public fun CharSequence.singleOrNull(): Char? {
    return if (length == 1) this[0] else null
}

/**
 * Returns the single character matching the given
 * [predicate], or `null` if character was not found or more than one character was found.
 */
public inline fun
CharSequence.singleOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found =
    false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single
            = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns a
 * subsequence of this char sequence with the first [n] characters removed.
 *
 * @throws
 * IllegalArgumentException if [n] is negative.
 *
 * @sample samples.text.Strings.drop
 */
public fun
CharSequence.drop(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return subSequence(n.coerceAtMost(length), length)
}

/**
 * Returns a string with the first [n] characters
 * removed.
 *
 * @throws IllegalArgumentException if [n] is negative.
 *
 * @sample
 * samples.text.Strings.drop
 */
public fun String.drop(n: Int): String {
    require(n >= 0) { "Requested character
    count $n is less than zero." }
    return substring(n.coerceAtMost(length))
}

/**
 * Returns a subsequence of
 * this char sequence with the last [n] characters removed.
 *
 * @throws IllegalArgumentException if [n] is
 * negative.
 *
 * @sample
 * samples.text.Strings.drop
 */
public fun CharSequence.dropLast(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return take((length -
    n).coerceAtLeast(0))
}

/**
 * Returns a string with the last [n] characters removed.
 *
 * @throws
 * IllegalArgumentException if [n] is negative.
 *
 * @sample
 * samples.text.Strings.drop
 */
public fun
String.dropLast(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return
    take((length - n).coerceAtLeast(0))
}

/**
 * Returns a subsequence of this char sequence containing all
 * characters except last characters that satisfy the given [predicate].
 *
 * @sample
 * samples.text.Strings.drop
 */

```

```

*^public inline fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return subSequence(0, index + 1)
    return ""
}
* Returns a string containing all characters except last characters that satisfy the given
[predicate].
* @sample samples.text.Strings.drop

*^public inline fun String.dropLastWhile(predicate:
(Char) -> Boolean): String {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return substring(0, index + 1)
    return ""
}
* Returns a subsequence of this char sequence containing all
characters except first characters that satisfy the given [predicate].
* @sample samples.text.Strings.drop

*^public inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices)
        if (!predicate(this[index]))
            return subSequence(index, length)
    return ""
}
* Returns a string containing all characters except first characters that satisfy the given [predicate].
* @sample samples.text.Strings.drop

*^public inline fun String.dropWhile(predicate:
(Char) -> Boolean): String {
    for (index in this.indices)
        if (!predicate(this[index]))
            return substring(index)
    return ""
}
* Returns a char sequence containing only those characters from the
original char sequence that match the given [predicate].
* @sample samples.text.Strings.filter

*^public
inline fun CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {
    return filterTo(StringBuilder(),
predicate)
}
* Returns a string containing only those characters from the original string that match the
given [predicate].
* @sample samples.text.Strings.filter

*^public inline fun String.filter(predicate: (Char) -
> Boolean): String {
    return filterTo(StringBuilder(), predicate).toString()
}
* Returns a char sequence
containing only those characters from the original char sequence that match the given [predicate].
* @param
[predicate] function that takes the index of a character
and the character itself
* and returns the result of predicate evaluation on the character.
* @sample
samples.collections.Collections.Filtering.filterIndexed

*^public inline fun CharSequence.filterIndexed(predicate:
(index: Int, Char) -> Boolean): CharSequence {
    return filterIndexedTo(StringBuilder(), predicate)
}
* Returns a string containing only those characters from the original string that match the given [predicate].
*
@param [predicate] function that takes the index of a character and the character itself
* and returns the result of
predicate evaluation on the character.
* @sample samples.collections.Collections.Filtering.filterIndexed

*^public inline fun String.filterIndexed(predicate: (index: Int, Char) -> Boolean): String {
    return
filterIndexedTo(StringBuilder(), predicate).toString()
}
* Appends all characters matching the given
[predicate] to the given [destination].
* @param [predicate] function that takes the
index of a character and the character itself
* and returns the result of predicate evaluation on the character.
* @sample samples.collections.Collections.Filtering.filterIndexedTo

*^public inline fun <C : Appendable>
CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    forEachIndexed {
index, element ->
        if (predicate(index, element)) destination.append(element)
    }
    return
destination
}
* Returns a char sequence containing only those characters from the original char sequence
that do not match the given [predicate].
* @sample samples.text.Strings.filterNot

*^public inline fun
CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {
    return filterNotTo(StringBuilder(),
predicate)
}
* Returns a string containing only those characters from the original string that do not match
the given [predicate].
* @sample samples.text.Strings.filterNot

*^public inline fun
String.filterNot(predicate: (Char) -> Boolean): String {
    return filterNotTo(StringBuilder(),
predicate).toString()
}
* Appends all characters not matching the given [predicate] to the given
[destination].
* @sample samples.collections.Collections.Filtering.filterTo

*^public inline fun <C :
Appendable> CharSequence.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this)
        if (!predicate(element)) destination.append(element)
    return destination
}
* Appends all characters
matching the given [predicate] to the given [destination].
* @sample
samples.collections.Collections.Filtering.filterTo

*^public inline fun <C : Appendable>
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (index in 0 until length) {
        val
element = get(index)
        if (predicate(element)) destination.append(element)
    }
    return
destination
}
* Returns a char sequence containing

```

characters of the original char sequence at the specified range of [indices].

```

public fun
CharSequence.slice(indices: IntRange): CharSequence {
    if (indices.isEmpty()) return ""
    return
    subSequence(indices)
}

```

* Returns a string containing characters of the original string at the specified range of [indices].

```

public fun String.slice(indices: IntRange): String {
    if (indices.isEmpty()) return ""
    return
    substring(indices)
}

```

* Returns a char sequence containing characters of the original char sequence at specified [indices].

```

public fun CharSequence.slice(indices: Iterable<Int>): CharSequence {
    val size =
    indices.collectionSizeOrDefault(10)
    if (size == 0) return ""
    val result = StringBuilder(size)
    for (i in
    indices) {
        result.append(get(i))
    }
    return result
}

```

* Returns a string containing characters of the original string at specified [indices].

```

@kotlin.internal.InlineOnly
public
inline fun String.slice(indices: Iterable<Int>): String {
    return (this as
    CharSequence).slice(indices).toString()
}

```

* Returns a subsequence of this char sequence containing the first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.take

```

```

public fun
CharSequence.take(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return
    subSequence(0, n.coerceAtMost(length))
}

```

* Returns a string containing the first [n] characters from this string, or the entire string if this string is shorter.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.take

```

```

public fun String.take(n: Int): String {
    require(n
    >= 0) { "Requested character count $n is less than zero." }
    return
    substring(0,
    n.coerceAtMost(length))
}

```

* Returns a subsequence of this char sequence containing the last [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.take

```

```

public fun
CharSequence.takeLast(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    val length = length
    return
    subSequence(length - n.coerceAtMost(length), length)
}

```

* Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.take

```

```

public fun String.takeLast(n: Int): String {
    require(n
    >= 0) { "Requested character count $n is less than zero." }
    val length = length
    return
    substring(length - n.coerceAtMost(length))
}

```

* Returns a subsequence of this char sequence containing last characters that satisfy the given [predicate].

```

@sample
samples.text.Strings.take

```

```

public inline fun
CharSequence.takeLastWhile(predicate: (Char) -> Boolean):
CharSequence {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return
            subSequence(index + 1, length)
        }
    }
    return
    subSequence(0, length)
}

```

* Returns a string containing last characters that satisfy the given [predicate].

```

@sample
samples.text.Strings.take

```

```

public inline fun
String.takeLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return
            substring(index + 1)
        }
    }
    return
    this
}

```

* Returns a subsequence of this char sequence containing the first characters that satisfy the given [predicate].

```

@sample
samples.text.Strings.take

```

```

public inline fun
CharSequence.takeWhile(predicate:
(Char) -> Boolean): CharSequence {
    for (index in 0 until length)
        if (!predicate(get(index))) {
            return
            subSequence(0, index)
        }
    return
    subSequence(0, length)
}

```

* Returns a string containing the first characters that satisfy the given [predicate].

```

@sample
samples.text.Strings.take

```

```

public inline fun
String.takeWhile(predicate: (Char) -> Boolean): String {
    for (index in 0 until length)
        if
        (!predicate(get(index))) {
            return
            substring(0, index)
        }
    return
    this
}

```

* Returns a char sequence with characters in reversed order.

```

public fun
CharSequence.reversed(): CharSequence {
    return
    StringBuilder(this).reverse()
}

```

* Returns a string with characters in reversed order.

```

@kotlin.internal.InlineOnly
public inline fun
String.reversed(): String {
    return (this as
    CharSequence).reversed().toString()
}

```

* Returns a [Map] containing key-value pairs provided by [transform] function applied to characters of the given char sequence.

```

If any of two
pairs would have the same key the last one gets added to the map.
The returned map preserves the entry

```

```

iteration order of the original char sequence.
 * @sample samples.text.Strings.associate
 * public inline fun
 <K, V> CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
   val capacity =
   mapCapacity(length).coerceAtLeast(16)
   return associateTo(LinkedHashMap<K, V>(capacity),
   transform)
 }
 * Returns a [Map] containing the characters from the given char sequence indexed by the
 key
 * returned from [keySelector] function applied to each character.
 * If any two characters would have the
 same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry
 iteration order of the original char sequence.
 * @sample samples.text.Strings.associateBy
 * public inline
 fun <K> CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {
   val capacity =
   mapCapacity(length).coerceAtLeast(16)
   return associateByTo(LinkedHashMap<K, Char>(capacity),
   keySelector)
 }
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by
 [keySelector] functions applied to characters of the given char sequence.
 * If any two characters would have
 the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the
 entry iteration order of the original char sequence.
 * @sample
 samples.text.Strings.associateByWithValueTransform
 * public inline fun <K, V>
 CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
   val capacity
 = mapCapacity(length).coerceAtLeast(16)
   return associateByTo(LinkedHashMap<K, V>(capacity),
   keySelector, valueTransform)
 }
 * Populates and returns the [destination] mutable map with key-value
 pairs,
 * where key is provided by the [keySelector] function applied to each character of the given char
 sequence
 * and value is the character itself.
 * If any two characters would have the same key returned by
 [keySelector] the last one gets added to the map.
 * @sample samples.text.Strings.associateByTo
 * public
 inline fun <K, M : MutableMap<in K, in Char>> CharSequence.associateByTo(destination: M, keySelector: (Char)
 -> K): M {
   for (element in this) {
     destination.put(keySelector(element), element)
   }
   return
 destination
 }
 * Populates and returns the [destination] mutable map with key-value pairs,
 * where key is
 provided by the [keySelector] function and
 * and value is provided by the [valueTransform] function applied to
 characters of the given char sequence.
 * If any two characters would have the same key returned by
 [keySelector] the last one gets added to the map.
 * @sample
 samples.text.Strings.associateByToWithValueTransform
 * public inline fun <K, V, M : MutableMap<in K, in V>> CharSequence.associateByTo(destination: M,
 keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
   for (element in this) {
     destination.put(keySelector(element), valueTransform(element))
   }
   return destination
 }
 * Populates
 and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each
 character of the given char sequence.
 * If any of two pairs would have the same key the last one gets added to
 the map.
 * @sample samples.text.Strings.associateTo
 * public inline fun <K, V, M : MutableMap<in K,
 in V>> CharSequence.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {
   for (element in this)
 {
     destination += transform(element)
   }
   return destination
 }
 * Returns a [Map] where keys are
 characters from the given char sequence and values are
 * produced by the [valueSelector]
 function applied to each character.
 * If any two characters are equal, the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original char sequence.
 * @sample
 samples.text.Strings.associateWith
 * @SinceKotlin("1.3")
 * public inline fun <V>
 CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {
   val result = LinkedHashMap<Char,
 V>(mapCapacity(length).coerceAtMost(128)).coerceAtLeast(16)
   return associateWithTo(result,
   valueSelector)
 }
 * Populates and returns the [destination] mutable map with key-value pairs for each
 character of the given char sequence,
 * where key is the character itself and value is provided by the
 [valueSelector] function applied to that key.
 * If any two characters are equal, the last one overwrites the
 former value in the map.
 * @sample samples.text.Strings.associateWithTo
 * @SinceKotlin("1.3")
 * public inline fun <V, M : MutableMap<in
 Char, in V>> CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
   for (element in
 this) {
     destination.put(element, valueSelector(element))
   }
   return destination
 }
 * Appends all

```

```

characters to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Char>>
CharSequence.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return
destination\n}\n\n/**\n * Returns a new [HashSet] of all characters.\n */\npublic fun CharSequence.toHashSet():
HashSet<Char> {\n return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))\n}\n\n/**\n *
Returns a [List] containing all characters.\n */\npublic fun CharSequence.toList(): List<Char> {\n return when
(length) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n *
Returns a new [MutableList] filled with all characters of this char sequence.\n
*/\npublic fun CharSequence.toMutableList(): MutableList<Char> {\n return
toCollection(ArrayList<Char>(length))\n}\n\n/**\n * Returns a [Set] of all characters.\n * \n * The returned set
preserves the element iteration order of the original char sequence.\n */\npublic fun CharSequence.toSet():
Set<Char> {\n return when (length) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else ->
toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))\n }\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each character of original char
sequence.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
CharSequence.flatMap(transform: (Char) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each character\n
* and its index in the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharSequence.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each character\n * and its index in the original char sequence, to the given
[destination].\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapIndexedTo(destination: C, transform:
(index: Int, Char) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each character of original char sequence, to
the given [destination].\n */\n\npublic inline fun <R, C : MutableCollection<in R>>
CharSequence.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {\n for (element in this) {\n val
list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups characters
of the original char sequence by the key returned by the given [keySelector] function\n * applied to each character
and returns a map where each group key is associated with a list of corresponding characters.\n * \n * The returned
map preserves the entry iteration order of the keys produced from the original char sequence.\n
* \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n\npublic inline fun <K>
CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each character of the original char sequence\n * by the key returned by the
given [keySelector] function applied to the character\n * and returns a map where each group key is associated with
a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n\npublic inline fun <K, V>
CharSequence.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
characters of the

```



```

* Applies the given [transform] function to each character in the original char sequence\n * and appends only the
non-null results to the given [destination].\n *\npublic inline fun <R : Any, C : MutableCollection<in R>>
CharSequence.mapNotNullTo(destination: C, transform: (Char) -> R?): C {\n  forEach { element ->
transform(element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies the given [transform]
function to each character of the original char sequence\n * and appends the results to the given [destination].\n
*\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C, transform: (Char) ->
R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Returns a
lazy [Iterable] that wraps each character of the original char sequence\n * into an [IndexedValue] containing the
index of that character and the character itself.\n *\npublic fun CharSequence.withIndex():
Iterable<IndexedValue<Char>>
{\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns `true` if all characters match the given
[predicate].\n *\n * @sample samples.collections.Collections.Aggregates.all\n *\npublic inline fun
CharSequence.all(predicate: (Char) -> Boolean): Boolean {\n  for (element in this) if (!predicate(element)) return
false\n  return true\n}\n\n/**\n * Returns `true` if char sequence has at least one character.\n *\n * @sample
samples.collections.Collections.Aggregates.any\n *\npublic fun CharSequence.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if at least one character matches the given [predicate].\n *\n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n *\npublic inline fun CharSequence.any(predicate:
(Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n/**\n * Returns the length of this char sequence.\n *\n * @kotlin.internal\npublic
inline fun CharSequence.count(): Int {\n  return length\n}\n\n/**\n * Returns the number of characters matching
the given [predicate].\n *\npublic inline fun CharSequence.count(predicate: (Char) -> Boolean): Int {\n  var count
= 0\n  for (element in this) if (predicate(element)) ++count\n  return count\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each
character.\n *\n * Returns the specified [initial] value if the char sequence is empty.\n *\n * @param [operation]
function that takes current accumulator value and a character, and calculates the next accumulator value.\n
*\npublic inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {\n  var accumulator =
initial\n  for (element in this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation]
from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n
*\n * Returns the specified [initial] value if the char sequence is empty.\n *\n * @param [operation] function that
takes the index of a character, current accumulator value\n * and the character itself, and calculates the next
accumulator value.\n *\npublic inline fun <R> CharSequence.foldIndexed(initial: R, operation: (index: Int, acc: R,
Char) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element in this) accumulator =
operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each character and current accumulator value.\n
*\n * Returns the specified [initial] value if the char sequence is empty.\n *\n * @param [operation] function that
takes a character and current accumulator value, and calculates the next accumulator
value.\n *\npublic inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {\n  var
index = lastIndex\n  var accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--),
accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each character with its index in the original char sequence and current
accumulator value.\n *\n * Returns the specified [initial] value if the char sequence is empty.\n *\n * @param
[operation] function that takes the index of a character, the character itself\n * and current accumulator value, and
calculates the next accumulator value.\n *\npublic inline fun <R> CharSequence.foldRightIndexed(initial: R,
operation: (index: Int, Char, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator = initial\n  while
(index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Performs the given [action] on each character.\n *\npublic
inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {\n  for (element in this)

```



```

action(element)\n}\n\n/**\n * Performs the given [action] on each character, providing sequential index with the
character.\n * @param [action] function that takes the index of a character and the character itself\n * and performs
the action on the character.\n */\npublic inline fun CharSequence.forEachIndexed(action: (index: Int, Char) -> Unit):
Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n@Deprecated("Use maxOrNull
instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic fun CharSequence.max(): Char? {\n    return
maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince
= "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>>
CharSequence.maxBy(selector: (Char) -> R): Char? {\n    return maxByOrNull(selector)\n}\n\n/**\n * Returns the
first character yielding the largest value of the given function or `null` if there are no characters.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharSequence.maxByOrNull(selector: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var
maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxV =
selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxV < v)
{\n            maxElem = e\n            maxV = v\n        }\n    }\n    return maxElem\n}\n\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each character
in the char sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOf(selector: (Char) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxV = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxV = maxOf(maxV, v)\n    }\n    return
maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun CharSequence.maxOf(selector: (Char) -> Float): Float {\n    if (isEmpty()) throw
NoSuchElementException()\n    var maxV = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        maxV = maxOf(maxV, v)\n    }\n    return maxV\n}\n\n/**\n * Returns the
largest value among all values produced by [selector] function\n * applied to each character in the char sequence.\n
* \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.maxOf(selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxV = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxV < v) {\n
            maxV = v\n        }\n    }\n    return
maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxV = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxV = maxOf(maxV, v)\n    }\n    return
maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there

```

```

are no characters.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOrNull(selector:
(Char) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.maxOrNull(selector: (Char) -> R): R? {\n    if
(isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each character in the char sequence.\n * \n * @throws NoSuchElementException if the char
sequence is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest character or `null` if there are no characters.\n *\n@SinceKotlin("1.4")\npublic fun
CharSequence.maxOrNull(): Char? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex)
{\n        val
e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun CharSequence.maxWith(comparator: Comparator<in Char>): Char?
{\n    return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first character having the largest value
according to the provided [comparator] or `null` if there are no characters.\n *\n@SinceKotlin("1.4")\npublic fun
CharSequence.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n    if (isEmpty()) return null\n    var
max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n
    }\n    return max\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun CharSequence.min():
Char? {\n    return minOrNull()\n}\n\n@Deprecated("Use minByOrNull instead.",
ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> CharSequence.minBy(selector: (Char) ->
R): Char? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first character yielding the smallest value of
the given function or `null` if there are no characters.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n *\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var
minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue =

```

```

selector(minElem)\n for (i in 1..lastIndex) {\n     val e = this[i]\n     val v = selector(e)\n     if (minValue > v)\n     {\n         minElem = e\n         minValue = v\n     }\n }\n return minElem\n}\n\n/**\n * Returns the smallest value among all values\n produced by [selector] function\n * applied to each character in the char sequence.\n * \n * If any of values produced\n by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the char\n sequence is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.minOf(selector: (Char) ->\n Double): Double {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue = minOf(minValue, v)\n     }\n     return\n     minValue\n }\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each character in the char sequence.\n * \n * If any of values produced\n by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the char\n sequence is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.minOf(selector: (Char) ->\n Float): Float {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in\n 1..lastIndex) {\n         val v = selector(this[i])\n         minValue = minOf(minValue, v)\n     }\n     return\n     minValue\n }\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun <R : Comparable<R>>\n CharSequence.minOf(selector: (Char) -> R): R {\n     if (isEmpty()) throw NoSuchElementException()\n     var\n     minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         if (minValue > v) {\n             minValue = v\n         }\n     }\n     return minValue\n }\n\n/**\n * Returns the smallest value among all values\n produced by [selector] function\n * applied to each character in the char sequence or `null` if there are no\n characters.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.minOrNull(selector:\n (Char) -> Double): Double? {\n     if (isEmpty()) return null\n     var minValue = selector(this[0])\n     for (i in\n 1..lastIndex) {\n         val v = selector(this[i])\n         minValue = minOf(minValue, v)\n     }\n     return minValue\n }\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * \n * applied to each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced\n by [selector] function is `NaN`, the returned result is `NaN`.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.minOrNull(selector:\n (Char) -> Float): Float? {\n     if (isEmpty()) return null\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex)\n {\n         val v = selector(this[i])\n         minValue = minOf(minValue, v)\n     }\n     return minValue\n }\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each character in the char\n sequence or `null` if there are no characters.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public\n inline fun <R : Comparable<R>> CharSequence.minOrNull(selector: (Char) -> R): R? {\n     if (isEmpty()) return\n     null\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         if\n (minValue > v) {\n             minValue = v\n         }\n     }\n     return minValue\n }\n\n/**\n * Returns the smallest value\n according to the provided [comparator]\n * among all values produced by [selector] function applied to each\n character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n
minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each character in the char sequence or
`null` if there are no characters.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n *
Returns the smallest character or `null` if there are no characters.\n
*/\n@SinceKotlin("1.4")\npublic fun CharSequence.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var
min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return
min\n}\n\n@Deprecated("Use minWithOrNull instead.")
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun CharSequence.minWith(comparator: Comparator<in Char>): Char? {\n
return minWithOrNull(comparator)\n}\n\n/**\n * Returns the first character having the smallest value according to
the provided [comparator] or `null` if there are no characters.\n
*/\n@SinceKotlin("1.4")\npublic fun
CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {\n    if (isEmpty()) return null\n    var
min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(min, e) > 0) min = e\n
}\n    return min\n}\n\n/**\n * Returns `true` if the char sequence has no characters.\n
*/\n * @sample
samples.collections.Collections.Aggregates.none\n */\npublic fun CharSequence.none(): Boolean {\n    return
isEmpty()\n}\n\n/**\n * Returns `true` if no characters match the given [predicate].\n
*/\n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
CharSequence.none(predicate: (Char) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Performs the given [action] on each character and returns the char sequence itself
afterwards.\n
*/\n@SinceKotlin("1.1")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) -> Unit): S
{\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each
character, providing sequential index with the character,\n * and returns the char sequence itself afterwards.\n
*/\n * @param [action] function that takes the index of
a character and the character itself\n * and performs the action on the character.\n
*/\n@SinceKotlin("1.4")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index: Int, Char) ->
Unit): S {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first
character and applying [operation] from left to right\n * to current accumulator value and each character.\n
*/\n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n
*/\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n
*/\n * @param [operation] function
that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n
*/\n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
CharSequence.reduce(operation: (acc: Char, Char) -> Char): Char {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty char sequence can't
be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator =
operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the
first character and applying [operation] from left to right\n * to current accumulator value and each character with its
index in the original char sequence.\n
*/\n * Throws an exception if this char sequence is empty. If the char sequence
can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is

```

```

empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value and the
character itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n * \npublic inline fun CharSequence.reduceIndexed(operation:
(index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw
UnsupportedOperationException("Empty
char sequence can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator
= operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first character and applying [operation] from left to right\n * to current accumulator value and each
character with its index in the original char sequence.\n * \n * Returns `null` if the char sequence is empty.\n * \n *
@param [operation] function that takes the index of a character, current accumulator value and the character itself,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n * \n@SinceKotlin("1.4")\npublic inline fun
CharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,
accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first
character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n *
Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes current accumulator
value and a character,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
* \n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation]
from right to left\n * to each character and current accumulator
value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an
expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes a character and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \npublic inline
fun CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index
< 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator =
get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to
left\n * to each character with its index in the original char sequence and current accumulator
value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an
expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n *
\n * @param [operation] function that takes the index of a character, the character itself and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n * \npublic inline fun
CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n var index =
lastIndex\n if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n
var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index),
accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last
character and applying [operation] from right to left\n * to
each character with its index in the original char sequence and current accumulator value.\n * \n * Returns `null` if
the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, the character
itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n@SinceKotlin("1.4")\npublic inline fun
CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n var index =

```

```

lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last character and applying [operation] from right to left\n * to each character
and current accumulator value.\n * \n * Returns `null` if the char sequence is empty.\n * \n *
@param [operation] function that takes a character and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharSequence.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each character and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes
current accumulator value and a character, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n */\n\n@SinceKotlin("1.4")\npublic inline fun <R>
CharSequence.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n  if (isEmpty()) return
listOf(initial)\n  val result = ArrayList<R>(length + 1).apply { add(initial) }\n  var accumulator = initial\n  for
(element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each character, its index in the original char sequence and current accumulator
value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R> CharSequence.runningFoldIndexed(initial: R, operation: (index:
Int, acc: R, Char) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(length +
1).apply { add(initial) }\n  var accumulator = initial\n  for (index in indices) {\n    accumulator =
operation(index, accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and a character, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n */\n\n@SinceKotlin("1.4")\npublic inline fun
CharSequence.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply { add(accumulator) }\n  for
(index in 1 until length) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each character, its index in the original char
sequence and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that
`acc` value passed
to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n *
\n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n */\n\n@SinceKotlin("1.4")\npublic inline fun
CharSequence.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply {

```



```

var sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char
sequence.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> ULong):
ULong {\n  var sum: ULong = 0.toULong()\n  for (element in this) {\n    sum += selector(element)\n  }\n
return
sum\n}\n\n/**\n * Splits this char sequence into a list of strings each not exceeding the given [size].\n * \n * The
last string in the resulting list may have fewer characters than the given [size].\n * \n * @param size the number of
elements to take in each string, must be positive and can be greater than the number of elements in this char
sequence.\n * \n * @sample samples.text.Strings.chunked\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.chunked(size: Int): List<String> {\n  return windowed(size, size, partialWindows =
true)\n}\n\n/**\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and
applies the given [transform] function to an each.\n * \n * @return list of results of the [transform] applied to an
each char sequence.\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid
only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a
snapshot of it.\n * The last char sequence may have fewer characters than the given [size].\n * \n * @param size the
number of elements to take in each char sequence, must be positive and can be greater than the number of elements
in this char sequence.\n * \n * @sample samples.text.Strings.chunkedTransform\n
*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R):
List<R> {\n  return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Splits this
char sequence into a sequence of strings each not exceeding the given [size].\n * \n * The last string in the resulting
sequence may have fewer characters than the given [size].\n * \n * @param size the number of elements to take in
each string, must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.chunked\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.chunkedSequence(size:
Int): Sequence<String> {\n  return chunkedSequence(size) { it.toString() }\n}\n\n/**\n * Splits this char sequence
into several char sequences each not exceeding the given [size]\n * and applies the given [transform] function to an
each.\n * \n * @return sequence of results of the [transform] applied to an each char sequence.\n * \n * Note that the
char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should
not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last char sequence may have
fewer characters than the given [size].\n * \n * @param size the number of elements to take in each char sequence,
must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample
samples.text.Strings.chunkedTransformToSequence\n *\n@SinceKotlin("1.2")\npublic fun <R>
CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {\n
return windowedSequence(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Splits the
original char sequence into pair of char sequences,\n * where *first* char sequence contains characters for which
[predicate] yielded `true`,\n * while *second* char sequence contains characters for which [predicate] yielded
`false`.\n * \n * @sample samples.text.Strings.partition\n *\npublic inline fun CharSequence.partition(predicate:
(Char) -> Boolean): Pair<CharSequence, CharSequence> {\n  val first = StringBuilder()\n  val second =
StringBuilder()\n  for (element in this) {\n    if (predicate(element)) {\n      first.append(element)\n    } else
{\n      second.append(element)\n    }\n  }\n  return Pair(first, second)\n}\n\n/**\n * Splits the original string
into pair of strings,\n * where *first* string contains characters for which [predicate] yielded `true`,\n * while
*second* string contains characters for which [predicate] yielded
`false`.\n * \n * @sample samples.text.Strings.partition\n *\npublic inline fun String.partition(predicate: (Char) ->
Boolean): Pair<String, String> {\n  val first = StringBuilder()\n  val second = StringBuilder()\n  for (element in
this) {\n    if (predicate(element)) {\n      first.append(element)\n    } else {\n

```



```

second.append(element)\n    }\n }\n return Pair(first.toString(), second.toString())\n}\n\n/**\n * Returns a list
of snapshots of the window of the given [size]\n * sliding along this char sequence with the given [step], where
each\n * snapshot is a string.\n * \n * Several last strings may have fewer characters than the given [size].\n * \n *
Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n *
@param size the number of elements to take in each window\n * @param step the number of elements to move the
window forward by on an each step, by default 1\n * @param partialWindows
controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial
windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.takeWindows\n
*\n@SinceKotlin("1.2")\npublic fun CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean =
false): List<String> {\n    return windowed(size, step, partialWindows) { it.toString() }\n}\n\n/**\n * Returns a list
of results of applying the given [transform] function to\n * an each char sequence representing a view over the
window of the given [size]\n * sliding along this char sequence with the given [step].\n * \n * Note that the char
sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not
store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last char sequences may
have fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and
can be greater than the number of elements in this char sequence.\n * @param size the number of elements to take
in each window\n * @param step the number of elements to move the window forward by on an each step, by
default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by
default `false` which means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.averageWindows\n *\n@SinceKotlin("1.2")\npublic fun <R>
CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R):
List<R> {\n    checkWindowSizeStep(size, step)\n    val thisSize = this.length\n    val resultCapacity = thisSize / step
+ if (thisSize % step == 0) 0 else 1\n    val result = ArrayList<R>(resultCapacity)\n    var index = 0\n    while (index
in 0 until thisSize) {\n        val end = index + size\n        val coercedEnd = if (end < 0 || end > thisSize) { if
(partialWindows)
thisSize else break } else end\n        result.add(transform(subSequence(index, coercedEnd)))\n        index += step\n
}\n    return result\n}\n\n/**\n * Returns a sequence of snapshots of the window of the given [size]\n * sliding along
this char sequence with the given [step], where each\n * snapshot is a string.\n * \n * Several last strings may have
fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the
number of elements in this char sequence.\n * @param size the number of elements to take in each window\n *
@param step the number of elements to move the window forward by on an each step, by default 1\n * @param
partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which
means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.takeWindows\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.windowedSequence(size: Int,
step: Int = 1, partialWindows: Boolean = false): Sequence<String> {\n    return windowedSequence(size, step,
partialWindows) { it.toString() }\n}\n\n/**\n * Returns a sequence of results of applying the given [transform]
function to\n * an each char sequence representing a view over the window of the given [size]\n * sliding along this
char sequence with the given [step].\n * \n * Note that the char sequence passed to the [transform] function is
ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way,
unless you made a snapshot of it.\n * Several last char sequences may have fewer characters than the given [size].\n
*\n * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n
*\n * @param size the number of elements to take in each window\n * @param step the number of elements to move
the window forward by on an each step, by default 1\n * @param partialWindows controls whether
or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be
preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n
*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1,
partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {\n

```

```

checkWindowSizeStep(size, step)\n    val windows = (if (partialWindows) indices else 0 until length - size + 1) step
step\n    return windows.asSequence().map { index ->\n        val end = index + size\n        val coercedEnd = if (end <
0 || end > length) length else end\n        transform(subSequence(index, coercedEnd))\n    }\n}\n\n/**\n * Returns a
list of pairs built from the characters of `this` and the [other] char sequences with the same index\n * The returned
list has length of the shortest char sequence.\n * \n * @sample samples.text.Strings.zip\n */\npublic infix fun
CharSequence.zip(other: CharSequence):
List<Pair<Char, Char>> {\n    return zip(other) { c1, c2 -> c1 to c2 }\n}\n\n/**\n * Returns a list of values built
from the characters of `this` and the [other] char sequences with the same index\n * using the provided [transform]
function applied to each pair of characters.\n * The returned list has length of the shortest char sequence.\n * \n *
@sample samples.text.Strings.zipWithTransform\n */\npublic inline fun <V> CharSequence.zip(other:
CharSequence, transform: (a: Char, b: Char) -> V): List<V> {\n    val length = minOf(this.length, other.length)\n
val list = ArrayList<V>(length)\n    for (i in 0 until length) {\n        list.add(transform(this[i], other[i]))\n    }\n
return list\n}\n\n/**\n * Returns a list of pairs of each two adjacent characters in this char sequence.\n * \n * The
returned list is empty if this char sequence contains less than two characters.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin("1.2")\npublic
fun CharSequence.zipWithNext(): List<Pair<Char, Char>> {\n    return zipWithNext { a, b -> a to b }\n}\n\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent
characters in this char sequence.\n * \n * The returned list is empty if this char sequence contains less than two
characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n */\n@SinceKotlin("1.2")\npublic inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R):
List<R> {\n    val size = length - 1\n    if (size < 1) return emptyList()\n    val result = ArrayList<R>(size)\n    for
(index in 0 until size) {\n        result.add(transform(this[index], this[index + 1]))\n    }\n    return result\n}\n\n/**\n *
Creates an [Iterable] instance that wraps the original char sequence returning its characters when being iterated.\n *
\n */\npublic fun CharSequence.asIterable(): Iterable<Char> {\n
if (this is String && isEmpty()) return emptyList()\n    return Iterable { this.iterator() }\n}\n\n/**\n * Creates a
[Sequence] instance that wraps the original char sequence returning its characters when being iterated.\n * \n */\npublic
fun CharSequence.asSequence(): Sequence<Char> {\n    if (this is String && isEmpty()) return emptySequence()\n    return
Sequence { this.iterator() }\n}\n\n"/>\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\nimport kotlin.contracts.contract\nimport kotlin.jvm.JvmName\n\n/**\n * Returns a copy of this string
converted to upper case using the rules of the default locale.\n * \n */\n@Deprecated("Use uppercase() instead.",
ReplaceWith("uppercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
expect fun String.toUpperCase(): String\n\n/**\n * Returns a copy of this string converted to upper case using
Unicode mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one
character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n
\n * \n * @sample samples.text.Strings.uppercase\n */\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.uppercase(): String\n\n/**\n * Returns a copy of this string converted to lower case using the rules of the
default locale.\n * \n */\n@Deprecated("Use lowercase() instead.",
ReplaceWith("lowercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.toLowerCase(): String\n\n/**\n * Returns a copy of this string converted to lower case using Unicode
mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one character
mapping,\n * thus the
length of the returned string can be different from the length of the original string.\n * \n * @sample
samples.text.Strings.lowercase\n */\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun

```



```

Returns a string having leading whitespace removed.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
String.trimStart(): String = (this as CharSequence).trimStart().toString()\n\n/**\n * Returns a sub sequence of this
char sequence having trailing whitespace removed.\n */\npublic fun CharSequence.trimEnd(): CharSequence =
trimEnd(Char::isWhitespace)\n\n/**\n * Returns a string having trailing whitespace removed.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun String.trimEnd(): String = (this as
CharSequence).trimEnd().toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded at
the beginning\n * to the specified [length] with the specified
character or space.\n * \n * @param length the desired string length.\n * @param padChar the character to pad
string with, if it has length less than the [length] specified. Space is used by default.\n * @return Returns a char
sequence of length at least [length] consisting of `this` char sequence prepended with [padChar] as many times\n *
as are necessary to reach that length.\n * @sample samples.text.Strings.padStart\n */\npublic fun
CharSequence.padStart(length: Int, padChar: Char = ' '): CharSequence {\n if (length < 0)\n throw
IllegalArgumentException("Desired length $length is less than zero.")\n if (length <= this.length)\n return
this.subSequence(0, this.length)\n\n val sb = StringBuilder(length)\n for (i in 1..(length - this.length))\n
sb.append(padChar)\n sb.append(this)\n return sb\n}\n\n/**\n * Pads the string to the specified [length] at the
beginning with the specified character or space.\n * \n * @param length the desired string
length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified. Space
is used by default.\n * @return Returns a string of length at least [length] consisting of `this` string prepended with
[padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padStart\n
*/\npublic fun String.padStart(length: Int, padChar: Char = ' '): String =\n (this as CharSequence).padStart(length,
padChar).toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded at the end\n * to
the specified [length] with the specified character or space.\n * \n * @param length the desired string length.\n *
@param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by
default.\n * @return Returns a char sequence of length at least [length] consisting of `this` char sequence appended
with [padChar] as many times\n * as are necessary to reach that
length.\n * @sample samples.text.Strings.padEnd\n */\npublic fun CharSequence.padEnd(length: Int, padChar:
Char = ' '): CharSequence {\n if (length < 0)\n throw IllegalArgumentException("Desired length $length is
less than zero.")\n if (length <= this.length)\n return this.subSequence(0, this.length)\n\n val sb =
StringBuilder(length)\n sb.append(this)\n for (i in 1..(length - this.length))\n sb.append(padChar)\n return
sb\n}\n\n/**\n * Pads the string to the specified [length] at the end with the specified character or space.\n * \n *
@param length the desired string length.\n * @param padChar the character to pad string with, if it has length less
than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length]
consisting of `this` string appended with [padChar] as many times\n * as are necessary to reach that length.\n *
@param length the desired string length.\n * @param padChar the character to pad string with, if it has length less
than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length]
consisting of `this` string appended with [padChar] as many times\n * as are necessary to reach that length.\n *
@sample samples.text.Strings.padEnd\n */\npublic fun String.padEnd(length:
Int, padChar: Char = ' '): String =\n (this as CharSequence).padEnd(length, padChar).toString()\n\n/**\n * Returns
`true` if this nullable char sequence is either `null` or empty.\n * \n * @sample
samples.text.Strings.stringOrNullEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence?.isNullOrEmpty(): Boolean {\n contract {\n returns(false) implies (this@isNullOrEmpty !=
null)\n }\n\n return this == null || this.length == 0\n}\n\n/**\n * Returns `true` if this char sequence is empty
(contains no characters).\n * \n * @sample samples.text.Strings.stringIsEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isEmpty(): Boolean = length == 0\n\n/**\n *
Returns `true` if this char sequence is not empty.\n * \n * @sample samples.text.Strings.stringIsNotEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotEmpty(): Boolean = length > 0\n\n//
implemented differently in JVM and JS\n//\npublic fun String.isBlank(): Boolean
= length() == 0 || all { it.isWhitespace() }\n\n/**\n * Returns `true` if this char sequence is not empty and contains
some characters except of whitespace characters.\n * \n * @sample samples.text.Strings.stringIsNotBlank\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotBlank(): Boolean = !isBlank()\n\n/**\n *
Returns `true` if this nullable char sequence is either `null` or empty or consists solely of whitespace characters.\n

```

```

*\n * @sample samples.text.Strings.stringOrNullOrNull\n *\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence?.isNullOrNull(): Boolean {\n    contract {\n        returns(false) implies (this@isNullOrNull !=
null)\n    }\n\n    return this == null || this.isBlank()\n}\n\n/**\n * Iterator for characters of the given char sequence.\n *\n@public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {\n    private var index = 0\n\n    public override fun nextChar(): Char = get(index++)\n\n    public override
fun hasNext(): Boolean = index < length\n}\n\n/** Returns the string if it is not `null`, or the empty string
otherwise. *\n@kotlin.internal.InlineOnly\npublic inline fun String?.orEmpty(): String = this ?: ""\n\n/**\n * Returns this char sequence if it's not empty\n * or the result of calling [defaultValue] function if the char sequence is
empty.\n *\n * @sample samples.text.Strings.stringIfEmpty\n *\n@kotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifEmpty(defaultValue: () ->
R): R where C : CharSequence, C : R =\n    if (isEmpty()) defaultValue() else this\n\n/**\n * Returns this char
sequence if it is not empty and doesn't consist solely of whitespace characters,\n * or the result of calling
[defaultValue] function otherwise.\n *\n * @sample samples.text.Strings.stringIfBlank\n *\n@kotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifBlank(defaultValue: () -> R):
R where C : CharSequence, C : R =\n    if (isBlank()) defaultValue()
else this\n\n/**\n * Returns the range of valid character indices for this char sequence.\n *\n@public val
CharSequence.indices: IntRange\n    get() = 0..length - 1\n\n/**\n * Returns the index of the last character in the
char sequence or -1 if it is empty.\n *\n@public val CharSequence.lastIndex: Int\n    get() = this.length - 1\n\n/**\n * Returns `true` if this CharSequence has Unicode surrogate pair at the specified [index].\n *\n@public fun
CharSequence.hasSurrogatePairAt(index: Int): Boolean {\n    return index in 0..length - 2\n        &&
this[index].isHighSurrogate()\n        && this[index + 1].isLowSurrogate()\n}\n\n/**\n * Returns a substring
specified by the given [range] of indices.\n *\n@public fun String.substring(range: IntRange): String =
substring(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence specified by
the given [range] of indices.\n *\n@public fun CharSequence.subSequence(range: IntRange): CharSequence
= subSequence(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence.\n *\n * This extension is chosen only for invocation with old-named parameters.\n * Replace parameter names with the
same as those of [CharSequence.subSequence].\n\n *\n@kotlin.internal.InlineOnly\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false
warning\n@Deprecated("Use parameters named startIndex and endIndex.", ReplaceWith("subSequence(startIndex
= start, endIndex = end)"))\npublic inline fun String.subSequence(start: Int, end: Int): CharSequence =
subSequence(start, end)\n\n/**\n * Returns a substring of chars from a range of this char sequence starting at the
[startIndex] and ending right before the [endIndex].\n *\n * @param startIndex the start index (inclusive).\n *
@param endIndex the end index (exclusive). If not specified, the length of the char sequence is used.\n\n *\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.substring(startIndex: Int, endIndex:
Int = length): String = subSequence(startIndex, endIndex).toString()\n\n/**\n * Returns a substring of chars at
indices from the specified [range] of this char sequence.\n *\n@public fun CharSequence.substring(range: IntRange):
String = subSequence(range.start, range.endInclusive + 1).toString()\n\n/**\n * Returns a substring before the first
occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which
defaults to the original string.\n *\n@public fun String.substringBefore(delimiter: Char, missingDelimiterValue:
String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else
substring(0, index)\n}\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does
not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n@public fun
String.substringBefore(delimiter: String, missingDelimiterValue: String
= this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else
substring(0, index)\n}\n\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does
not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n@public fun
String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index =
indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index + 1, length)\n}\n\n/**\n *

```

Returns a substring after the first occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
`public fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this): String {
 val index = indexOf(delimiter)
 return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)
}`

* Returns a substring before the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
`public fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else substring(0, index)
}`

Returns a substring before the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
`public fun String.substringBeforeLast(delimiter: String, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else substring(0, index)
}`

* Returns a substring after the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
`public fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else substring(index + 1, length)
}`

* Returns a substring after the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
`public fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)
}`

* Returns a char sequence with content of this char sequence where its part at the given range is replaced with the [replacement] char sequence.
 * @param startIndex the index of the first character to be replaced.
 * @param endIndex the index of the first character after the replacement to keep in the string.
`public fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): CharSequence {
 if (endIndex < startIndex) throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")
 val sb = StringBuilder()
 sb.appendRange(this, 0, startIndex)
 sb.append(replacement)
 sb.appendRange(this, endIndex, length)
 return sb
}`

* Replaces the part of the string at the given range with the [replacement] char sequence.
 * @param startIndex the index of the first character to be replaced.
 * @param endIndex the index of the first character after the replacement to keep in the string.
`@kotlin.internal.InlineOnly
public inline fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): String =
 (this as CharSequence).replaceRange(startIndex, endIndex, replacement).toString()`

* Returns a char sequence with content of this char sequence where its part at the given [range] is replaced with the [replacement] char sequence.
 * The end index of the [range] is included in the part to be replaced.
`public fun CharSequence.replaceRange(range: IntRange, replacement: CharSequence): CharSequence =
 replaceRange(range.start, range.endInclusive + 1, replacement)`

* Replace the part of string at the given [range] with the [replacement] string.
 * The end index of the [range] is included in the part to be replaced.
`@kotlin.internal.InlineOnly
public inline fun String.replaceRange(range: IntRange, replacement: CharSequence): String =
 (this as CharSequence).replaceRange(range, replacement).toString()`

* Returns a char sequence with content of this char sequence where its part at the given range is removed.
 * @param startIndex the index of the first character to be removed.
 * @param endIndex the index of the first character after the removed part to keep in the string.
 * [endIndex] is not included in the removed part.
`public fun CharSequence.removeRange(startIndex: Int, endIndex: Int): CharSequence {
 if (endIndex < startIndex) throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")
 if (endIndex == startIndex) return this.subSequence(0, length)
 val sb = StringBuilder(length - (endIndex - startIndex))
 sb.appendRange(this, 0, startIndex)
 sb.appendRange(this, endIndex, length)
 return sb
}`

* Removes the part of a string at a given range.
 * @param startIndex the index of the first character

to be removed.\n * @param endIndex the index of the first character after the removed part to keep in the string.\n * \n * [endIndex] is not included in the removed part.\n * \n * @kotlin.internal.InlineOnly\npublic inline fun String.removeRange(startIndex: Int, endIndex: Int): String =\n (this as CharSequence).removeRange(startIndex, endIndex).toString()\n\n\n * Returns a char sequence with content of this char sequence where its part at the given [range] is removed.\n * \n * The end index of the [range] is included in the removed part.\n * \n * @public fun CharSequence.removeRange(range: IntRange): CharSequence = removeRange(range.start, range.endInclusive + 1)\n\n\n * Removes the part of a string at the given [range].\n * \n * The end index of the [range] is included in the removed part.\n * \n * @kotlin.internal.InlineOnly\npublic inline fun String.removeRange(range: IntRange): String =\n (this as CharSequence).removeRange(range).toString()\n\n\n * If this char sequence starts with the given [prefix], returns a new char sequence\n * with the prefix removed. Otherwise, returns a new char sequence with the same characters.\n * \n * @public fun CharSequence.removePrefix(prefix: CharSequence): CharSequence {\n if (startsWith(prefix)) {\n return subSequence(prefix.length, length)\n }\n return subSequence(0, length)\n}\n\n\n * If this string starts with the given [prefix], returns a copy of this string\n * with the prefix removed. Otherwise, returns this string.\n * \n * @public fun String.removePrefix(prefix: CharSequence): String {\n if (startsWith(prefix)) {\n return substring(prefix.length)\n }\n return this\n}\n\n\n * If this char sequence ends with the given [suffix], returns a new char sequence\n * with the suffix removed. Otherwise, returns a new char sequence with the same characters.\n * \n * @public fun CharSequence.removeSuffix(suffix: CharSequence): CharSequence {\n if (endsWith(suffix)) {\n return subSequence(0, length - suffix.length)\n }\n return subSequence(0, length)\n}\n\n\n * If this string ends with the given [suffix], returns a copy of this string\n * with the suffix removed. Otherwise, returns this string.\n * \n * @public fun String.removeSuffix(suffix: CharSequence): String {\n if (endsWith(suffix)) {\n return substring(0, length - suffix.length)\n }\n return this\n}\n\n\n * When this char sequence starts with the given [prefix] and ends with the given [suffix],\n * returns a new char sequence having both the given [prefix] and [suffix] removed.\n * \n * Otherwise returns a new char sequence with the same characters.\n * \n * @public fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence {\n if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n return subSequence(prefix.length, length - suffix.length)\n }\n return subSequence(0, length)\n}\n\n\n * Removes from a string both the given [prefix] and [suffix] if and only if\n * it starts with the [prefix] and ends with the [suffix].\n * \n * Otherwise returns this string unchanged.\n * \n * @public fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {\n if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n return substring(prefix.length, length - suffix.length)\n }\n return this\n}\n\n\n * When this char sequence starts with and ends with the given [delimiter],\n * returns a new char sequence having this [delimiter] removed both from the start and end.\n * \n * Otherwise returns a new char sequence with the same characters.\n * \n * @public fun CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence = removeSurrounding(delimiter, delimiter)\n\n\n\n * Removes the given [delimiter] string from both the start and the end of this string\n * if and only if it starts with and ends with the [delimiter].\n * \n * Otherwise returns this string unchanged.\n * \n * @public fun String.removeSurrounding(delimiter: CharSequence): String = removeSurrounding(delimiter, delimiter)\n\n\n\n * Replace part of string before the first occurrence of given delimiter with the [replacement] string.\n * \n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n * \n * @public fun String.replaceBefore(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n\n\n * Replace part of string before the first occurrence of given delimiter with the [replacement] string.\n * \n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n * \n * @public fun String.replaceBefore(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n\n\n * Replace part

of string after the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfter(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n\n/**\n * Replace part of string after the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfter(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)\n}\n\n/**\n * Replace part of string after the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfterLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)\n}\n\n/**\n * Replace part of string after the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfterLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n\n/**\n * Replace part of string before the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBeforeLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n/**\n * Replace part of string before the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBeforeLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n// public fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific\n// public fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific\n\n/**\n * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression\n * with the given [replacement].\n * The [replacement] can consist of any combination of literal text and \$-substitutions. To treat the replacement string\n * literally escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)\n\n/**\n * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression\n * with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a\n * replacement for that match.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> CharSequence): String =\n regex.replace(this, transform)\n\n/**\n * Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.\n * @param replacement A replacement expression that can include substitutions. See [Regex.replaceFirst] for details.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String = regex.replaceFirst(this, replacement)\n\n/**\n * Returns a copy of this string having its first character replaced with the result of the specified [transform],\n * or the original string if it's empty.\n * @param transform function that takes the first character and returns the result of the transform applied to the character.\n * @sample samples.text.Strings.replaceFirstChar\n */\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@OptIn(kotlin.experimental.Exper


```

imentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@JvmName("replaceFirstCharWithC
har")\n@kotlin.internal.InlineOnly\npublic
    inline fun String.replaceFirstChar(transform: (Char) -> Char): String {\n    return if (isEmpty())
transform(this[0]) + substring(1) else this\n}\n\n/**\n * Returns a copy of this string having its first character
replaced with the result of the specified [transform],\n * or the original string if it's empty.\n * @param
transform function that takes the first character and returns the result of the transform applied to the character.\n *
*\n * @sample samples.text.Strings.replaceFirstChar\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@OptIn(kotlin.experimental.Exper
imentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@JvmName("replaceFirstCharWithC
harSequence")\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceFirstChar(transform: (Char) ->
CharSequence): String {\n    return if (isEmpty()) transform(this[0]).toString() + substring(1) else
this\n}\n\n/**\n * Returns `true` if this char sequence matches the given regular expression.\n
*\n@kotlin.internal.InlineOnly\npublic inline infix fun CharSequence.matches(regex: Regex): Boolean =
regex.matches(this)\n\n/**\n * Implementation of [regionMatches] for CharSequences.\n * Invoked when it's
already known that arguments are not Strings, so that no additional type checks are performed.\n *\ninternal fun
CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase:
Boolean): Boolean {\n    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset >
other.length - length)) {\n        return false\n    }\n    for (index in 0 until length) {\n        if (!this[thisOffset +
index].equals(other[otherOffset + index], ignoreCase))\n            return false\n    }\n    return true\n}\n\n/**\n *
Returns `true` if this char sequence starts with the specified character.\n *\npublic fun
CharSequence.startsWith(char:
    Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 && this[0].equals(char, ignoreCase)\n\n/**\n *
Returns `true` if this char sequence ends with the specified character.\n *\npublic fun CharSequence.endsWith(char:
    Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 && this[lastIndex].equals(char,
ignoreCase)\n\n/**\n * Returns `true` if this char sequence starts with the specified prefix.\n *\npublic fun
CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {\n    if (!ignoreCase &&
this is String && prefix is String)\n        return this.startsWith(prefix)\n    else\n        return regionMatchesImpl(0,
prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this char sequence starting at the
specified offset [startIndex] starts with the specified prefix.\n *\npublic fun CharSequence.startsWith(prefix:
    CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n
        if (!ignoreCase && this is String && prefix is String)\n            return this.startsWith(prefix, startIndex)\n        else\n
            return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this char
sequence ends with the specified suffix.\n *\npublic fun CharSequence.endsWith(suffix: CharSequence,
ignoreCase: Boolean = false): Boolean {\n    if (!ignoreCase && this is String && suffix is String)\n        return
this.endsWith(suffix)\n    else\n        return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length,
ignoreCase)\n}\n\n\n// common prefix and suffix\n\n/**\n * Returns the longest string `prefix` such that this char
sequence and [other] char sequence both start with this prefix,\n * taking care not to split surrogate pairs.\n * If this
and [other] have no common prefix, returns the empty string.\n * @param ignoreCase `true` to ignore character
case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonPrefixWith\n
*\npublic fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n
    val shortestLength = minOf(this.length, other.length)\n    var i = 0\n    while (i < shortestLength &&
this[i].equals(other[i], ignoreCase = ignoreCase)) {\n        i++\n    }\n    if (this.hasSurrogatePairAt(i - 1) ||
other.hasSurrogatePairAt(i - 1)) {\n        i--\n    }\n    return subSequence(0, i).toString()\n}\n\n/**\n * Returns the
longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,\n * taking
care not to split surrogate pairs.\n * If this and [other] have no common suffix, returns the empty string.\n\n *
@param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample
samples.text.Strings.commonSuffixWith\n *\npublic fun CharSequence.commonSuffixWith(other: CharSequence,

```

```

ignoreCase: Boolean = false): String {\n  val thisLength = this.length\n
  val otherLength = other.length\n  val shortestLength = minOf(thisLength, otherLength)\n\n  var i = 0\n  while
(i < shortestLength && this[thisLength - i - 1].equals(other[otherLength - i - 1], ignoreCase = ignoreCase)) {\n
  i++\n  }\n  if (this.hasSurrogatePairAt(thisLength - i - 1) || other.hasSurrogatePairAt(otherLength - i - 1)) {\n
  i--\n  }\n  return subSequence(thisLength - i, thisLength).toString()\n}\n\n// indexOfAny()\n\n/**\n * Finds the
index of the first occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified
[startIndex] and optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when
matching a character. By default `false`.\n *\n * @return An index of the first occurrence of matched character from
[chars] or -1 if none of [chars] are found.\n *\n */\n\npublic fun CharSequence.indexOfAny(chars: CharArray,
startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n  if
(!ignoreCase && chars.size == 1 && this is String) {\n    val char = chars.single()\n    return
nativeIndexOf(char, startIndex)\n  }\n\n  for (index in startIndex.coerceAtLeast(0)..lastIndex) {\n    val
charAtIndex = get(index)\n    if (chars.any { it.equals(charAtIndex, ignoreCase) })\n      return index\n  }\n
return -1\n}\n\n/**\n * Finds the index of the last occurrence of any of the specified [chars] in this char sequence,\n
* starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param startIndex The index of
character to start searching at. The search proceeds backward toward the beginning of the string.\n *\n * @param
ignoreCase `true` to ignore character case when matching a character. By default `false`.\n *\n * @return An index of
the last occurrence of matched character from [chars] or -1 if none of [chars] are found.\n *\n */\n\npublic fun
CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase:
Boolean = false): Int {\n  if (!ignoreCase && chars.size == 1 && this is String) {\n    val char =
chars.single()\n    return nativeLastIndexOf(char, startIndex)\n  }\n\n  for (index in startIndex.coerceAtMost(lastIndex)
downTo 0) {\n    val charAtIndex = get(index)\n    if (chars.any { it.equals(charAtIndex, ignoreCase) })\n
return index\n  }\n\n  return -1\n}\n\nprivate fun CharSequence.indexOf(other: CharSequence, startIndex: Int,
endIndex: Int, ignoreCase: Boolean, last: Boolean = false): Int {\n  val indices = if (!last)\n
startIndex.coerceAtLeast(0)..endIndex.coerceAtMost(length)\n  else\n    startIndex.coerceAtMost(lastIndex)
downTo endIndex.coerceAtLeast(0)\n\n  if (this is String && other is String) { // smart cast\n    for (index in
indices) {\n      if (other.regionMatches(0, this, index, other.length, ignoreCase))\n        return index\n
    }\n  } else {\n    for (index in indices) {\n
      if (other.regionMatchesImpl(0, this, index, other.length, ignoreCase))\n        return index\n    }\n  }\n
return -1\n}\n\nprivate fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int, ignoreCase:
Boolean, last: Boolean): Pair<Int, String>? {\n  if (!ignoreCase && strings.size == 1) {\n    val string =
strings.single()\n    val index = if (!last) indexOf(string, startIndex) else lastIndexOf(string, startIndex)\n
return if (index < 0) null else index to string\n  }\n\n  val indices = if (!last) startIndex.coerceAtLeast(0)..length
else startIndex.coerceAtMost(lastIndex) downTo 0\n\n  if (this is String) {\n    for (index in indices) {\n
val matchingString = strings.firstOrNull { it.regionMatches(0, this, index, it.length, ignoreCase) }\n    if
(matchingString != null)\n      return index to matchingString\n    }\n  } else {\n    for (index in indices)
{\n      val matchingString
= strings.firstOrNull { it.regionMatchesImpl(0, this, index, it.length, ignoreCase) }\n      if (matchingString !=
null)\n        return index to matchingString\n    }\n  }\n\n  return null\n}\n\n/**\n * Finds the first
occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and
optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when matching a string. By
default `false`.\n *\n * @return A pair of an index of the first occurrence of matched string from [strings] and the
string matched\n * or `null` if none of [strings] are found.\n *\n * To avoid ambiguous results when strings in [strings]
have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each
position the first element in [strings]\n * that matches this string at that position.\n *\n */\n\npublic fun
CharSequence.findAnyOf(strings: Collection<String>, startIndex:
Int = 0, ignoreCase: Boolean = false): Pair<Int, String>? =\n  findAnyOf(strings, startIndex, ignoreCase, last =
false)\n\n/**\n * Finds the last occurrence of any of the specified [strings] in this char sequence,\n * starting from

```

the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the last occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the end toward the beginning of this string, and finds at each position the first element in [strings] that matches this string at that position.\n */\npublic fun CharSequence.findLastAnyOf(strings:

Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =\n

findAnyOf(strings, startIndex, ignoreCase, last = true)\n */\n * Finds the index of the first occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the beginning to the end of this string, and finds at each position the first element in [strings] that matches this string at that position.\n */\npublic fun CharSequence.indexOfAny(strings: Collection<String>, startIndex: Int = 0,

ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase,

last = false)?.first ?: -1\n */\n * Finds the index of the last occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of matched string from [strings] or -1 if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the end toward the beginning of this string, and finds at each position the first element in [strings] that matches this string at that position.\n */\npublic fun CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int =

lastIndex, ignoreCase: Boolean = false): Int =\n findAnyOf(strings,

startIndex, ignoreCase, last = true)?.first ?: -1\n */\n // indexOf\n */\n * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the first occurrence of [char] or -1 if none is found.\n */\npublic fun CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n indexOfAny(charArrayOf(char), startIndex, ignoreCase)\n else\n nativeIndexOf(char, startIndex)\n }\n */\n * Returns the index within this char sequence of the first occurrence of the specified [string], starting from the specified [startIndex].\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of [string] or -1 if none is found.\n

* @sample samples.text.Strings.indexOf\n */\npublic fun CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n indexOf(string, startIndex, length, ignoreCase)\n else\n nativeIndexOf(string, startIndex)\n }\n */\n * Returns the index within this char sequence of the last occurrence of the specified character, starting from the specified [startIndex].\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the last occurrence of [char] or -1 if none is found.\n */\npublic fun

CharSequence.lastIndexOf(char: Char, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)\n else\n nativeLastIndexOf(char, startIndex)\n }\n */\n * Returns the index within this char sequence of the last occurrence of the specified [string], starting from the specified [startIndex].\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of [string] or -1 if none is found.\n */\npublic fun

```

CharSequence.lastIndexOf(string: String, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n  return if
(ignoreCase || this !is String)\n    indexOf(string, startIndex, 0, ignoreCase, last = true)\n  else\n  nativeLastIndexOf(string, startIndex)\n}\n\n/**\n * Returns `true` if this char sequence contains the specified [other]
sequence of characters as a substring.\n * @param ignoreCase `true` to ignore character
case when comparing strings. By default `false`.\n
*/\n@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic operator fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =\n  if (other is String)\n  indexOf(other, ignoreCase = ignoreCase) >= 0\n  else\n  indexOf(other, 0, length, ignoreCase) >=
0\n\n\n/**\n * Returns `true` if this char sequence contains the specified character [char].\n * @param
ignoreCase `true` to ignore character case when comparing characters. By default `false`.\n
*/\n@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic operator fun CharSequence.contains(char:
Char, ignoreCase: Boolean = false): Boolean =\n  indexOf(char, ignoreCase = ignoreCase) >= 0\n\n\n/**\n * Returns
`true` if this char sequence contains at least one match of the specified regular expression [regex].\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun CharSequence.contains(regex: Regex): Boolean =
regex.containsMatchIn(this)\n\n\n//
rangesDelimitedBy\n\nprivate class DelimitedRangesSequence(\n  private val input: CharSequence,\n  private
val startIndex: Int,\n  private val limit: Int,\n  private val getNextMatch: CharSequence.(currentIndex: Int) ->
Pair<Int, Int>? ) : Sequence<IntRange> {\n  override fun iterator(): Iterator<IntRange> = object :
Iterator<IntRange> {\n    var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n    var
currentStartIndex: Int = startIndex.coerceIn(0, input.length)\n    var nextSearchIndex: Int = currentStartIndex\n
var nextItem: IntRange? = null\n    var counter: Int = 0\n    private fun calcNext() {\n      if
(nextSearchIndex < 0) {\n        nextState = 0\n        nextItem = null\n      } else {\n        if (limit >
0 && ++counter >= limit || nextSearchIndex > input.length) {\n          nextItem =
currentStartIndex..input.lastIndex\n          nextSearchIndex
= -1\n        } else {\n          val match = input.getNextMatch(nextSearchIndex)\n          if (match ==
null) {\n            nextItem = currentStartIndex..input.lastIndex\n            nextSearchIndex = -1\n
          } else {\n            val (index, length) = match\n            nextItem = currentStartIndex until index\n
            currentStartIndex = index + length\n            nextSearchIndex = currentStartIndex + if (length ==
0) 1 else 0\n          }\n          nextState = 1\n        }\n      }\n    }\n    override fun next():
IntRange {\n      if (nextState == -1)\n        calcNext()\n      if (nextState == 0)\n        throw
NoSuchElementException()\n      val result = nextItem as IntRange\n      // Clean next to avoid keeping
reference on yielded instance\n      nextItem = null\n
      nextState = -1\n      return result\n    }\n  }\n  override fun hasNext(): Boolean {\n    if (nextState
== -1)\n      calcNext()\n    return nextState == 1\n  }\n}\n\n\n/**\n * Returns a sequence of index
ranges of substrings in this char sequence around occurrences of the specified [delimiters].\n * @param
delimiters One or more characters to be used as delimiters.\n * @param startIndex The index to start searching
delimiters from.\n * No range having its start value less than [startIndex] is returned.\n * [startIndex] is coerced to
be non-negative and not greater than length of this string.\n * @param ignoreCase `true` to ignore character case
when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return. Zero
by default means no limit is set.\n */\nprivate fun CharSequence.rangesDelimitedBy(delimiters: CharArray,
startIndex: Int = 0, ignoreCase: Boolean = false, limit: Int = 0):
Sequence<IntRange> {\n  requireNonNegativeLimit(limit)\n  return DelimitedRangesSequence(this,
startIndex, limit, { currentIndex ->\n    indexOfAny(delimiters, currentIndex, ignoreCase = ignoreCase).let { if (it
< 0) null else it to 1 } })\n}\n\n\n/**\n * Returns a sequence of index ranges of substrings in this char sequence
around occurrences of the specified [delimiters].\n * @param delimiters One or more strings to be used as
delimiters.\n * @param startIndex The index to start searching delimiters from.\n * No range having its start value
less than [startIndex] is returned.\n * [startIndex] is coerced to be non-negative and not greater than length of this
string.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *

```

```

@param limit The maximum number of substrings to return. Zero by default means no limit is set.
To avoid ambiguous results when strings in [delimiters] have characters in common, this method
proceeds from the beginning to the end of this string, and finds at each position the first element in
[delimiters] that matches this string at that position.
private fun
CharSequence.rangesDelimitedBy(delimiters: Array<out String>, startIndex: Int = 0, ignoreCase: Boolean = false,
limit: Int = 0): Sequence<IntRange> {
    requireNonNegativeLimit(limit)
    val delimitersList = delimiters.asList()
    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->
        findAnyOf(delimitersList, currentIndex, ignoreCase = ignoreCase, last = false)?.let { it.first to it.second.length }
    })
}

internal fun requireNonNegativeLimit(limit: Int) = require(limit >= 0) { "Limit must be non-
negative, but was $limit" }

// split
/** Splits this char sequence to a sequence of strings around
occurrences of the specified [delimiters].
@param delimiters One or more strings to be used as delimiters.
@param ignoreCase `true` to ignore character
case when matching a delimiter. By default `false`.
@param limit The maximum number of substrings to return.
Zero by default means no limit is set.
To avoid ambiguous results when strings in [delimiters] have
characters in common, this method proceeds from the beginning to the end of this string, and finds at each
position the first element in [delimiters] that matches this string at that position.
public fun
CharSequence.splitToSequence(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0):
Sequence<String> =
    rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }
/** Splits this char sequence to a list of strings around occurrences of the specified [delimiters].
@param delimiters One or more strings to be used as delimiters.
@param ignoreCase `true` to ignore character
case when matching a delimiter. By default `false`.
@param limit The maximum number of substrings to return.
Zero by default means no limit is set.
To avoid ambiguous results when strings in [delimiters] have
characters in common, this method proceeds from the beginning to the end of this string, and matches at each
position the first element in [delimiters] that is equal to a delimiter in this instance at that position.
public fun
CharSequence.split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
    if (delimiters.size == 1) {
        val delimiter = delimiters[0]
        if (!delimiter.isEmpty()) {
            return split(delimiter, ignoreCase, limit)
        }
    }
    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}

/** Splits this char sequence to a sequence of
strings around occurrences of the specified [delimiters].
@param delimiters One or more characters to be
used as delimiters.
@param ignoreCase `true` to ignore character
case when matching a delimiter. By default `false`.
@param limit The maximum
number of substrings to return.
public fun
CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean = false, limit:
Int = 0): Sequence<String> =
    rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map {
        substring(it) }
/** Splits this char sequence to a list of strings around occurrences of the specified
[delimiters].
@param delimiters One or more characters to be used as delimiters.
@param ignoreCase
`true` to ignore character
case when matching a delimiter. By default `false`.
@param limit The maximum
number of substrings to return.
public fun
CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean =
false, limit: Int = 0): List<String> {
    if (delimiters.size == 1) {
        return split(delimiters[0].toString(),
ignoreCase, limit)
    }
    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit =
limit).asIterable().map
{ substring(it) }
}

/** Splits this char sequence to a list of strings around occurrences of the specified
[delimiter].
This is specialized version of split which receives single non-empty delimiter and offers better
performance.
@param delimiter String used as delimiter.
@param ignoreCase `true` to ignore character
case when matching a delimiter. By default `false`.
@param limit The maximum number of substrings to
return.
private fun
CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {
    requireNonNegativeLimit(limit)
    var currentOffset = 0
    var nextIndex = indexOf(delimiter, currentOffset,
ignoreCase)
    if (nextIndex == -1 || limit == 1) {
        return listOf(this.toString())
    }
    val isLimited =
limit > 0
    val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)
    do {

```

```

result.add(substring(currentOffset, nextIndex))\n    currentOffset
= nextIndex + delimiter.length\n    // Do not search for next occurrence if we're reaching limit\n    if (isLimited
&& result.size == limit - 1) break\n    nextIndex = indexOf(delimiter, currentOffset, ignoreCase)\n    } while
(nextIndex != -1)\n\n result.add(substring(currentOffset, length))\n return result\n}\n\n/**\n * Splits this char
sequence to a list of strings around matches of the given regular expression.\n *\n * @param limit Non-negative
value specifying the maximum number of substrings to return.\n * Zero by default means no limit is set.\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.split(regex: Regex, limit: Int = 0): List<String> =
regex.split(this, limit)\n\n/**\n * Splits this char sequence to a sequence of strings around matches of the given
regular expression.\n *\n * @param limit Non-negative value specifying the maximum number of substrings to
return.\n * Zero by default means no limit is set.\n * @sample samples.text.Strings.splitToSequence\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun CharSequence.splitToSequence(regex: Regex, limit: Int = 0): Sequence<String> =
regex.splitToSequence(this, limit)\n\n/**\n * Splits this char sequence to a sequence of lines delimited by any of the
following character sequences: CRLF, LF or CR.\n *\n * The lines returned do not include terminating line
separators.\n */\n\npublic fun CharSequence.lineSequence(): Sequence<String> = splitToSequence("\\r\\n", "\\n",
"\\r")\n\n/**\n * Splits this char sequence to a list of lines delimited by any of the following character sequences:
CRLF, LF or CR.\n *\n * The lines returned do not include terminating line separators.\n */\n\npublic fun
CharSequence.lines(): List<String> = lineSequence().toList()\n\n/**\n * Returns `true` if the contents of this char
sequence are equal to the contents of the specified [other],\n * i.e. both char sequences contain
the same number of the same characters in the same order.\n *\n * @sample samples.text.Strings.contentEquals\n
*/\n\n@SinceKotlin("1.5")\n\npublic expect infix fun CharSequence?.contentEquals(other: CharSequence?):
Boolean\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified
[other], optionally ignoring case difference.\n *\n * @param ignoreCase `true` to ignore character case when
comparing contents.\n *\n * @sample samples.text.Strings.contentEquals\n */\n\n@SinceKotlin("1.5")\n\npublic
expect fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean\n\n\ninternal fun
CharSequence?.contentEqualsIgnoreCaseImpl(other: CharSequence?): Boolean {\n    if (this is String && other is
String) {\n        return this.equals(other, ignoreCase = true)\n    }\n    if (this === other) return true\n    if (this ==
null || other == null || this.length != other.length) return false\n    for (i in 0 until length) {\n
        if (!this[i].equals(other[i], ignoreCase = true)) {\n            return false\n        }\n    }\n    return true\n}\n\n\ninternal
fun CharSequence?.contentEqualsImpl(other: CharSequence?): Boolean {\n    if (this is String && other is String)
{\n        return this == other\n    }\n    if (this === other) return true\n    if (this == null || other == null || this.length
!= other.length) return false\n    for (i in 0 until length) {\n        if (this[i] != other[i]) {\n            return false\n
        }\n    }\n    return true\n}\n\n\n/**\n * Returns `true` if the content of this string is equal to the word `true`, `false`
if it is equal to `false`,\n * and throws an exception otherwise.\n *\n * There is also a lenient version of the
function available on nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n
* @sample samples.text.Strings.toBooleanStrict\n */\n\n@SinceKotlin("1.5")\n\npublic fun String.toBooleanStrict():
Boolean = when (this)
{\n    "true" -> true\n    "false" -> false\n    else -> throw IllegalArgumentException("The string doesn't represent
a boolean value: $this")\n}\n\n\n/**\n * Returns `true` if the content of this string is equal to the word `true`, `false`
if it is equal to `false`,\n * and `null` otherwise.\n *\n * There is also a lenient version of the function available on
nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n * @sample
samples.text.Strings.toBooleanStrictOrNull\n */\n\n@SinceKotlin("1.5")\n\npublic fun
String.toBooleanStrictOrNull(): Boolean? = when (this) {\n    "true" -> true\n    "false" -> false\n    else ->
null\n},"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n//
Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic

```

```

value class UByteArray@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ByteArray) :
Collection<UByte> {\n\n /** Creates a new array of the specified [size], with all elements initialized to zero. *\n
public constructor(size: Int) : this(ByteArray(size))\n\n /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n * \n * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n * \n public
operator fun get(index: Int): UByte = storage[index].toUByte()\n\n /**\n * Sets the element at the given [index]
to the given [value]. This method can be called using the index operator.\n * \n * If the [index] is out of bounds
of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is
unspecified.\n
*\n public operator fun set(index: Int, value: UByte) {\n storage[index] = value.toByte()\n }\n\n /**
Returns the number of elements in the array. *\n public override val size: Int get() = storage.size\n\n /** Creates
an iterator over the elements of the array. *\n public override operator fun iterator():
kotlin.collections.Iterator<UByte> = Iterator(storage)\n\n @Suppress("DEPRECATION_ERROR")\n private
class Iterator(private val array: ByteArray) : UByteIterator() {\n private var index = 0\n override fun
hasNext() = index < array.size\n override fun nextUByte() = if (index < array.size) array[index++].toUByte()
else throw NoSuchElementException(index.toString())\n }\n\n override fun contains(element: UByte): Boolean
{\n // TODO: Eliminate this check after KT-30016 gets fixed.\n // Currently JS BE does not generate
special bridge method for this method.\n @Suppress("USELESS_CAST")\n if ((element
as Any?) !is UByte) return false\n return storage.contains(element.toByte())\n }\n\n override fun
containsAll(elements: Collection<UByte>): Boolean {\n return (elements as Collection<*>).all { it is UByte
&& storage.contains(it.toByte()) }\n }\n\n override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n
* Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n * \n * The function [init] is called for each array element sequentially starting from the first one.\n * It
should return the value for an array element given its index.\n
*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray(size: Int, init: (Int) -> UByte): UByteArray {\n return UByteArray(ByteArray(size) { index ->
init(index).toByte()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ubyteArrayOf(vararg
elements: UByte): UByteArray = elements\n"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n * \n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class
UIntArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: IntArray) : Collection<UInt>
{\n\n /** Creates a new array of the specified [size], with all elements initialized to zero. *\n public
constructor(size: Int) : this(IntArray(size))\n\n /**\n * Returns the array element at the given [index]. This
method can be called using the index operator.\n * \n * If the [index] is out of bounds of this array, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n * \n public
operator fun get(index: Int): UInt = storage[index].toInt()\n\n /**\n * Sets the element at the given [index] to
the given [value]. This method can be called using the index operator.\n * \n * If the [index] is out of bounds
of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n public operator fun set(index: Int, value: UInt) {\n storage[index] = value.toInt()\n }\n\n /** Returns
the number of elements in the array. *\n public override val size: Int get() = storage.size\n\n /** Creates an
iterator over the elements of the array. *\n public override operator fun iterator(): kotlin.collections.Iterator<UInt>
= Iterator(storage)\n\n @Suppress("DEPRECATION_ERROR")\n private class Iterator(private val array:
IntArray) : UIntIterator() {\n private var index = 0\n override fun hasNext() = index < array.size\n
override fun nextUInt() = if (index < array.size)

```

```

array[index++].toUInt() else throw NoSuchElementException(index.toString())\n } \n\n override fun
contains(element: UInt): Boolean {\n // TODO: Eliminate this check after KT-30016 gets fixed.\n //
Currently JS BE does not generate special bridge method for this method.\n
@Suppress("USELESS_CAST")\n if ((element as Any?) !is UInt) return false\n\n return
storage.contains(element.toInt())\n } \n\n override fun containsAll(elements: Collection<UInt>): Boolean {\n
return (elements as Collection<*>).all { it is UInt && storage.contains(it.toInt()) } \n } \n\n override fun
isEmpty(): Boolean = this.storage.size == 0\n\n/**\n * Creates a new array of the specified [size], where each
element is calculated by calling the specified\n * [init] function.\n * \n * The function [init] is called for each array
element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {\n return UIntArray(IntArray(size) { index ->
init(index).toInt()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
uintArrayOf(vararg elements: UInt): UIntArray = elements\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class ULongArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: LongArray) :
Collection<ULong> {\n\n /** Creates a new array of the specified [size], with all elements initialized to zero. */\n
public constructor(size:
Int) : this(LongArray(size))\n\n /**\n * Returns the array element at the given [index]. This method can be
called using the index operator.\n * \n * If the [index] is out of bounds of this array, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n * \n public
operator fun get(index: Int): ULong = storage[index].toULong()\n\n /**\n * Sets the element at the given
[index] to the given [value]. This method can be called using the index operator.\n * \n * If the [index] is out of
bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is
unspecified.\n * \n public operator fun set(index: Int, value: ULong) {\n storage[index] = value.toLong()\n
}\n\n /** Returns the number of elements in the array. */\n public override val size: Int get() = storage.size\n\n
/** Creates an iterator over the elements of the array. */\n
public override operator fun iterator(): kotlin.collections.Iterator<ULong> = Iterator(storage)\n\n
@Suppress("DEPRECATION_ERROR")\n private class Iterator(private val array: LongArray) : ULongIterator()\n
{\n private var index = 0\n override fun hasNext() = index < array.size\n override fun nextULong() = if
(index < array.size) array[index++].toULong() else throw NoSuchElementException(index.toString())\n } \n\n
override fun contains(element: ULong): Boolean {\n // TODO: Eliminate this check after KT-30016 gets
fixed.\n // Currently JS BE does not generate special bridge method for this method.\n
@Suppress("USELESS_CAST")\n if ((element as Any?) !is ULong) return false\n\n return
storage.contains(element.toLong())\n } \n\n override fun containsAll(elements: Collection<ULong>): Boolean
{\n return (elements as Collection<*>).all { it is ULong && storage.contains(it.toLong()) } \n } \n\n override
fun
isEmpty(): Boolean = this.storage.size == 0\n\n/**\n * Creates a new array of the specified [size], where each
element is calculated by calling the specified\n * [init] function.\n * \n * The function [init] is called for each array
element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray(size: Int, init: (Int) -> ULong): ULongArray {\n return ULongArray(LongArray(size) { index ->
init(index).toLong()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ulongArrayOf(vararg elements: ULong): ULongArray = elements\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license

```


*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UByteArray.component1(): UByte {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If
 the size of this array is less than 1, throws an [IndexOutOfBoundsException]
 except in Kotlin/JS\n * where the behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UShortArray.component1(): UShort {\n return get(0)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n *
 If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UIntArray.component2(): UInt {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If
 the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 ULongArray.component2(): ULong {\n return get(1)\n}\n\n/**\n * Returns 2nd *element*
 from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in
 Kotlin/JS\n * where the behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UByteArray.component2(): UByte {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n *
 If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UShortArray.component2(): UShort {\n return get(1)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n *
 If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
 operator fun UIntArray.component3(): UInt {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the
 array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n *
 * where the behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 ULongArray.component3(): ULong {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n *
 If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UByteArray.component3(): UByte {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n *
 If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
 behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline operator fun UShortArray.component3(): UShort {\n return get(2)\n}\n\n/**\n * Returns 4th *element*
 from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in
 Kotlin/JS\n * where the behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
 UIntArray.component4(): UInt {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If
 the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
 is unspecified.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline operator fun ULongArray.component4(): ULong {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from
 the array.\n * \n * If the size of this array is less than 4, throws
 an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun

UByteArray.component4(): UByte { \n return get(3)\n}\n\n/** \n * Returns 4th *element* from the array.\n * \n *
If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
```

UShortArray.component4(): UShort { \n return get(3)\n}\n\n/** \n * Returns 5th *element* from the array.\n * \n *
If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
```

UIntArray.component5(): UInt { \n return get(4)\n}\n\n/** \n * Returns
5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
```

ULongArray.component5(): ULong { \n return get(4)\n}\n\n/** \n * Returns 5th *element* from the array.\n * \n *
If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
```

UByteArray.component5(): UByte { \n return get(4)\n}\n\n/** \n * Returns 5th *element* from the array.\n * \n *
If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
```

inline operator fun UShortArray.component5(): UShort { \n return get(4)\n}\n\n/** \n * Returns an element at the
given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n *
@sample samples.collections.Collections.Elements.elementAt\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.elementAt(index: Int):
```

UInt\n\n/** \n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.elementAt(index: Int):
```

ULong\n\n/** \n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect
```

fun UByteArray.elementAt(index: Int): UByte\n\n/** \n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample

```
samples.collections.Collections.Elements.elementAt\n
```

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int):
```

UShort\n\n/** \n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
[index] is out of bounds of this array.\n * \n * @sample

```
samples.collections.Collections.Elements.elementAtOrElse\n
```

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt { \n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/** \n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds

```
of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
```

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong { \n return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/** \n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample

```
samples.collections.Collections.Elements.elementAtOrElse\n
```

```
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```



```

(ULong) -> Boolean): ULong? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching
the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.first(): UInt {\n    return storage.first().toUInt()\n}\n\n/**\n * Returns first element.\n *
@throws [NoSuchElementException] if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(): ULong {\n    return storage.first().toULong()\n}\n\n/**\n * Returns first element.\n * @throws
[NoSuchElementException] if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(): UByte {\n    return storage.first().toUByte()\n}\n\n/**\n * Returns first element.\n * @throws
[NoSuchElementException] if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(): UShort {\n    return storage.first().toUShort()\n}\n\n/**\n * Returns the first element matching
the given [predicate].\n * @throws [NoSuchElementException] if no
such element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(predicate: (UInt) -> Boolean): UInt {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(predicate: (ULong) -> Boolean): ULong {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.first(predicate: (UByte) -> Boolean): UByte {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(predicate: (UShort) -> Boolean): UShort {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.firstOrNull(): UInt? {\n    return
if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.firstOrNull(): ULong? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first
element, or `null` if the array is empty.\n * \n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.firstOrNull(): UByte? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first

```

```

element, or `null` if the array is empty.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.firstOrNull(): UShort? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first
element matching the given [predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {\n    for (element in this) if (predicate(element)) return
element\n    return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {\n    for (element in this) if (predicate(element))
return element\n    return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if
element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {\n    for (element in this) if (predicate(element))
return element\n    return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if
element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {\n    for (element in this) if (predicate(element))
return
element\n    return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.getOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this
array.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.getOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.getOrNull(index: Int): UInt? {\n
return if (index
>= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if
the [index] is out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.getOrNull(index: Int):
ULong? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.getOrNull(index: Int): UByte?
{\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.getOrNull\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.getOrNull(index: Int): UShort? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOf(element: UInt): Int {\n    return storage.indexOf(element.toInt())\n}\n\n/**\n * Returns first
index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOf(element: ULong): Int {\n    return storage.indexOf(element.toLong())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOf(element: UByte): Int {\n    return storage.indexOf(element.toByte())\n}\n\n/**\n *
Returns first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOf(element: UShort): Int {\n    return storage.indexOf(element.toShort())\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {\n    return storage.indexOfFirst { predicate(it.toInt())
}\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toULong()) }\n}\n\n/**\n *
Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {\n
    return storage.indexOfLast { predicate(it.toInt()) }\n}\n\n/**\n * Returns index of the last element matching the
given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(): UInt {\n    return storage.last().toInt()\n}\n\n/**\n * Returns the last element.\n * \n * @throws

```

```

NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(): UByte {\n    return storage.last().toUByte()\n}\n\n/**\n * Returns the last element.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.last(): UShort {\n    return storage.last().toUShort()\n}\n\n/**\n * Returns the last element matching
the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(predicate: (UInt) -> Boolean): UInt {\n    for (index in this.indices.reversed()) {\n        val element =
this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array
contains no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given
[predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(predicate: (UByte) -> Boolean): UByte {\n    for (index in this.indices.reversed()) {\n        val element =
this[index]\n        if (predicate(element)) return
element\n    }\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(predicate: (UShort) -> Boolean): UShort {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns last index
of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.lastIndexOf(element: UInt): Int {\n    return storage.lastIndexOf(element.toInt())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.lastIndexOf(element: UByte): Int {\n    return storage.lastIndexOf(element.toByte())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.lastIndexOf(element: UShort): Int {\n    return storage.lastIndexOf(element.toShort())\n}\n\n/**\n *

```


Returns

the last element, or `null` if the array is empty.

```
*\n @sample samples.collections.Collections.Elements.last\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.lastOrNull(): UInt? {\n     return\n     if (isEmpty()) null else this[size - 1]\n }\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.lastOrNull(): ULong? {\n     return\n     if (isEmpty()) null else this[size - 1]\n }\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UByteArray.lastOrNull(): UByte? {\n     return\n     if (isEmpty()) null else this[size - 1]\n }\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public\n     fun UShortArray.lastOrNull(): UShort? {\n         return\n         if (isEmpty()) null else this[size - 1]\n     }\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n     samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {\n         for (index in this.indices.reversed()) {\n             val\n             element = this[index]\n             if (predicate(element)) return element\n         }\n         return null\n     }\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n     samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {\n         for (index in this.indices.reversed())\n         {\n             val\n             element = this[index]\n             if (predicate(element)) return element\n         }\n         return null\n     }\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n     samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {\n         for (index in this.indices.reversed()) {\n             val\n             element = this[index]\n             if (predicate(element)) return element\n         }\n         return null\n     }\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n     samples.collections.Collections.Elements.last\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {\n         for (index in this.indices.reversed())\n         {\n             val\n             element = this[index]\n             if (predicate(element)) return element\n         }\n         return null\n     }\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     UIntArray.random(): UInt {\n         return\n         random(Random)\n     }\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     ULongArray.random(): ULong {\n         return\n         random(Random)\n     }\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n     UByteArray.random(): UByte {\n         return\n         random(Random)\n     }\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun\n     UIntArray.random(random: Random): UInt {\n         if (isEmpty())\n             throw\n             NoSuchElementException("Array is empty.")\n         return
```

```

get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.random(random: Random):
ULong {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n
* Returns a random element from this array using the specified source of randomness.\n * \n * @throws
NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.random(random: Random):
UByte {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.random(random: Random):
UShort {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.randomOrNull(): UInt? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.randomOrNull(): ULong? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.randomOrNull(): UByte? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.randomOrNull(): UShort?
{\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array using the specified
source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun UIntArray.randomOrNull(random: Random): UInt? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun ULongArray.randomOrNull(random: Random): ULong? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public
fun UByteArray.randomOrNull(random: Random): UByte? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun UShortArray.randomOrNull(random: Random): UShort? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(): UInt {\n return storage.single().toUInt()\n}\n\n/**\n * Returns the single element, or throws an
exception if the array is empty or has more than one element.\n

```

```

*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.single(): ULong {\n    return storage.single().toULong()\n}\n\n/**\n * Returns the single
element, or throws an exception if the array is empty or has more than one element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(): UByte {\n    return storage.single().toUByte()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(): UShort {\n    return storage.single().toUShort()\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(predicate: (UInt) -> Boolean): UInt {\n
    var single: UInt? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if
(found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single =
element\n                found = true\n            }\n        }\n        if (!found) throw NoSuchElementException("Array contains no
element matching the predicate.")\n    }\n    @Suppress("UNCHECKED_CAST")\n    return single as UInt\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one
matching element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(predicate: (ULong) -> Boolean): ULong {\n    var single: ULong? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n                single = element\n                found = true\n            }\n        }\n        if (!found) throw NoSuchElementException("Array
contains no element matching the predicate.")\n    }\n    @Suppress("UNCHECKED_CAST")\n    return single as
ULong\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or
more than one matching element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(predicate: (UByte) -> Boolean): UByte {\n    var single: UByte? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n                single = element\n                found = true\n            }\n        }\n        if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    @Suppress("UNCHECKED_CAST")\n    return single as UByte\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one
matching element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(predicate: (UShort) -> Boolean): UShort {\n    var single: UShort? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n                single = element\n                found = true\n            }\n        }\n        if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    @Suppress("UNCHECKED_CAST")\n    return single as UShort\n}\n\n/**\n * Returns single element, or `null` if
the array is empty or has more than one element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {\n
return if (size == 1) this[0]
else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull(): ULong? {\n
return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic fun
UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element,
or `null` if the array is empty or has more than one element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull(): UShort? {\n
return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element matching the given [predicate], or\n
`null` if element was not found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\n
inline fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n  var single: UInt? = null\n  var found\n
= false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single =\n
element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the\n
single element matching the given [predicate], or `null` if element was not found or more than one element was\n
found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline\n
fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n  var single: ULong? = null\n  var\n
found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single\n
= element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the\n
single element matching the given\n
[predicate], or `null` if element was not found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n
UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n  var single: UByte? = null\n  var found =\n
false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single =\n
element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the\n
single element matching the given [predicate], or `null` if element was not found or more than one element was\n
found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline\n
fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n  var single: UShort? = null\n  var\n
found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n\n
single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n
*\n * @throws IllegalArgumentException if [n] is\n
negative.\n
*\n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size -\n
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n
*\n * @throws\n
IllegalArgumentException if [n] is negative.\n
*\n * @sample\n
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size -\n
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n
*\n * @throws\n
IllegalArgumentException if [n]\n
is negative.\n
*\n * @sample\n
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size -\n
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n
*\n * @throws\n
IllegalArgumentException if [n] is negative.\n
*\n * @sample\n
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size -\n
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n
*\n * @throws\n
IllegalArgumentException\n
if [n] is negative.\n
*\n * @sample\n
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.dropLast(n: Int): List<UInt> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n  return take((size -\n
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n
*\n * @throws

```

```

IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.dropLast(n: Int):
List<ULong> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UByteArray.dropLast(n: Int): List<UByte> {\n    require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n    return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.dropLast(n: Int):
List<UShort> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropLastWhile(predicate: (UInt) -> Boolean):
List<UInt> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index
+ 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements
that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    for (index in lastIndex downTo 0)
{\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropLastWhile(predicate:
(UByte) -> Boolean): List<UByte> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    for (index in lastIndex downTo
0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return
emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    var yielding = false\n    val list =
ArrayList<UInt>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    var yielding = false\n    val list =
ArrayList<ULong>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except

```

```

first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {\n  var yielding = false\n  val list =
ArrayList<UByte>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))\n      list.add(item)\n    yielding = true\n  }\n  return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {\n  var yielding = false\n  val list =
ArrayList<UShort>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding =
true\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {\n  return filterTo(ArrayList<UInt>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {\n  return filterTo(ArrayList<ULong>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {\n  return filterTo(ArrayList<UByte>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {\n  return filterTo(ArrayList<UShort>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {\n  return
filterIndexedTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filterIndexed(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n  return
filterIndexedTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation
on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n  return
filterIndexedTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample

```

```

samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n  return
filterIndexedTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate]
to the given [destination].\n
 * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) -> Boolean): C
{\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n
return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <C : MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int,
ULong) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) ->
Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the
index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n *
@sample samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) ->
Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Returns a list containing all elements not
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {\n  return
filterNotTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {\n  return filterNotTo(ArrayList<ULong>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {\n  return filterNotTo(ArrayList<UByte>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {\n  return
filterNotTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements not matching the given [predicate]
to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterNotTo(destination:
C, predicate: (ULong) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n
return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given
[destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean):
C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n *
Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {\n for (element
in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {\n for
(element in this) if (predicate(element))
destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate]
to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n *
Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n

```



```

* Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):\nList<UShort> {\n    if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive\n+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):\nList<UInt> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =\nArrayList<UInt>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: Iterable<Int>):\nList<ULong> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size ==\n0) return emptyList()\n    val list = ArrayList<ULong>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: Iterable<Int>):\nList<UByte> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =\nArrayList<UByte>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: Iterable<Int>):\nList<UShort> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =\nArrayList<UShort>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns an array\n * containing elements of this array at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices:\nCollection<Int>): UIntArray {\n    return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array\n * containing elements of this array at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices:\nCollection<Int>): ULongArray {\n    return ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array\n * containing elements of this array at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices:\nCollection<Int>): UByteArray {\n    return UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array\n * containing elements of this array at specified [indices].\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sliceArray(indices:\nCollection<Int>): UShortArray {\n    return UShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in\n * the specified [indices] range.\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nUIntArray.sliceArray(indices: IntRange): UIntArray {\n    return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices: IntRange):\nULongArray {\n    return ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing\n * elements at indices in the specified [indices] range.\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices: IntRange):\nUByteArray {\n    return UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing\n * elements at indices in the specified [indices] range.\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic\nfun UShortArray.sliceArray(indices: IntRange): UShortArray {\n    return\nUShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * \n * @sample\n * samples.collections.Collections.Transformations.take\n *\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.take(n: Int): List<UInt> {\n

```

```

require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<UInt>(n)\n for
(item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.take(n: Int): List<ULong> {\n require(n >= 0) { \"Requested element count $n is less than
zero.\" } \n if (n == 0) return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n
var count = 0\n val list = ArrayList<ULong>(n)\n for (item in this) {\n list.add(item)\n if (++count ==
n)\n break\n }\n return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.take(n: Int): List<UByte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<UByte>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.take(n: Int): List<UShort> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<UShort>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.takeLast(n: Int): List<UInt> {\n require(n >= 0) { \"Requested element count $n is less than zero.\"
}\n if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return
listOf(this[size - 1])\n val list = ArrayList<UInt>(n)\n for (index in size - n until size)\n list.add(this[index])\n
return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.takeLast(n: Int): List<ULong>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<ULong>(n)\n for (index in size - n
until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n *
\n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.takeLast(n: Int): List<UByte>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UByte>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n
*\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.takeLast(n: Int): List<UShort>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UShort>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n

```

```

Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n  for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n      return drop(index + 1)\n  }\n } return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun
ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n  for (index in lastIndex
downTo 0) {\n  if (!predicate(this[index])) {\n      return drop(index + 1)\n  }\n }\n } return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n  for (index in lastIndex
downTo 0) {\n  if (!predicate(this[index])) {\n      return drop(index + 1)\n  }\n }\n } return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun
UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n  for (index in lastIndex
downTo 0) {\n  if (!predicate(this[index])) {\n      return drop(index + 1)\n  }\n }\n } return
toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {\n  val list = ArrayList<UInt>()\n  for (item in
this) {\n  if (!predicate(item))\n      break\n  list.add(item)\n }\n } return list\n}\n\n/**\n * Returns a list
containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun
ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {\n  val list =
ArrayList<ULong>()\n  for (item in this) {\n  if (!predicate(item))\n      break\n  list.add(item)\n }\n }
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {\n  val list = ArrayList<UByte>()\n  for
(item in this) {\n  if (!predicate(item))\n      break\n  list.add(item)\n }\n } return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun
UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {\n  val list =
ArrayList<UShort>()\n  for (item in this) {\n  if (!predicate(item))\n      break\n  list.add(item)\n }\n }
return list\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reverse(): Unit {\n  storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(): Unit {\n  storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(): Unit {\n  storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements of the array in the
 * specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is
 * less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if
 * [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range
 * (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws
 * IndexOutOfBoundsException if [fromIndex] is
 * less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if
 * [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

/**
 * Reverses elements
 * of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to
 * reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws
 * IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public
fun UIntArray.reversed(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public
fun ULongArray.reversed(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.reversed(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns
 * a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
UShortArray.reversed(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reversedArray(): UIntArray {
    return UIntArray(storage.reversedArray())
}

/**
 * Returns an
 * array
 * with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reversedArray(): ULongArray {
    return ULongArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reversedArray(): UByteArray {
    return UByteArray(storage.reversedArray())
}

/**
 * Returns
 * an array
 * with elements of this array in reversed order.
 */

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reversedArray(): UShortArray {\n    return UShortArray(storage.reversedArray())\n}\n\n/**\n *
Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles
elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.shuffle(): Unit {\n    shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-
place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]
instance as the source of randomness.\n
* \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(random: Random): Unit
{\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j]
= copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance
as the source of randomness.\n
* \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(random: Random):
Unit {\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n
* \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(random: Random):
Unit {\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n
* \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(random: Random):
Unit {\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their
natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements
in the array in-place descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(): Unit {\n    if
(size > 1)
{\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their
natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.sortDescending(): Unit {\n    if (size > 1) {\n        sort()\n        reverse()\n    }\n}\n\n/**\n * Sorts
elements in the array in-place descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(): Unit {\n
if (size > 1) {\n    sort()\n    reverse()\n }\n}\n\n/**\n * Returns a list of all elements sorted according to their
natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sorted():
List<UInt> {\n    return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted
according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sorted():

```

```

List<ULong> {\n    return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted
according to their natural sort order.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.sorted(): List<UByte> {\n    return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all
elements sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sorted(): List<UShort> {\n
return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns an array with all elements of this array sorted
according to their natural sort order.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.sortedArray(): UIntArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sort()
}\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.sortedArray(): ULongArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sort()
}\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedArray(): UByteArray {\n
if (isEmpty()) return this\n    return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements
of this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArray(): UShortArray
{\n    if (isEmpty()) return this\n    return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all
elements of this array sorted descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedArrayDescending():
UIntArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply
{\n        sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to
their natural sort order.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sortedArrayDescending(): ULongArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply {\n
sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to
their natural sort order.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.sortedArrayDescending(): UByteArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply {\n
sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to
their natural sort order.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.sortedArrayDescending(): UShortArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply {\n
sortDescending() }\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n
*\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedDescending(): List<UInt>
{\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending
according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order
relative to each other after sorting.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sortedDescending(): List<ULong> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n
*\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UByteArray.sortedDescending(): List<UByte> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n
*\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedDescending():
List<UShort> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns an array of type [ByteArray],
which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.asByteArray(): ByteArray {\n    return storage\n}\n\n/**\n * Returns an array of type [IntArray], which

```

is a view of this array where each element is a signed reinterpretation
of the corresponding element of this array.

```

*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.asIntArray(): IntArray {\n    return storage\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic expect fun UIntArray.asList():
List<UInt>\n\n/**\n * Returns a [List] that wraps the original array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic expect fun ULongArray.asList():
List<ULong>\n\n/**\n * Returns a [List] that wraps the original array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic expect fun UByteArray.asList():
List<UByte>\n\n/**\n * Returns a [List] that wraps the original array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic expect fun UShortArray.asList():
List<UShort>\n\n/**\n * Returns an array of type [LongArray], which is a view of this array where each element is
a signed reinterpretation
of the corresponding element of this array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.asLongArray(): LongArray {\n    return storage\n}\n\n/**\n * Returns an array of type [ShortArray],
which is a view of this array where each element is a signed reinterpretation
of the corresponding element of this
array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.asShortArray(): ShortArray {\n    return storage\n}\n\n/**\n * Returns an array of type [UByteArray],
which is a view of this array where each element is an unsigned reinterpretation
of the corresponding element of
this array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline
fun ByteArray.asUByteArray(): UByteArray {\n    return UByteArray(this)\n}\n\n/**\n * Returns an array of type
[UIntArray], which is a view of this array where
each element is an unsigned reinterpretation
of the corresponding element of this array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
IntArray.asUIntArray(): UIntArray {\n    return UIntArray(this)\n}\n\n/**\n * Returns an array of type
[ULongArray], which is a view of this array where each element is an unsigned reinterpretation
of the
corresponding element of this array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
LongArray.asULongArray(): ULongArray {\n    return ULongArray(this)\n}\n\n/**\n * Returns an array of type
[UShortArray], which is a view of this array where each element is an unsigned reinterpretation
of the
corresponding element of this array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.asUShortArray(): UShortArray {\n    return UShortArray(this)\n}\n\n/**\n * Returns `true`
if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same
elements in the same order.\n
*^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince =
"1.4")@ExperimentalUnsignedTypes\npublic infix fun UIntArray.contentEquals(other: UIntArray): Boolean {\n
return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to
one another,\n * i.e. contain the same number of the same elements in the same order.\n
*^@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince =
"1.4")@ExperimentalUnsignedTypes\npublic infix fun ULongArray.contentEquals(other: ULongArray):
Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain
the same number of the same elements in the same order.\n
*^@Deprecated("Use Kotlin compiler 1.4 to avoid
deprecation warning.")@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince =
"1.4")@ExperimentalUnsignedTypes\npublic infix fun UByteArray.contentEquals(other: UByteArray): Boolean

```

```

{\n  return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal
to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray.contentEquals(other: UShortArray):
Boolean {\n  return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic
infix fun UIntArray?.contentEquals(other: UIntArray?): Boolean {\n  return
this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally*
equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray?.contentEquals(other:
ULongArray?): Boolean {\n  return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the
two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements
in the same order.\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic infix fun
UByteArray?.contentEquals(other: UByteArray?): Boolean {\n  return
this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally*
equal to one another,\n * i.e. contain the same number
of the same elements in the same order.\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic infix
fun UShortArray?.contentHashCode(): Int {\n  return
this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as
if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentHashCode(): Int {\n  return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentHashCode(): Int {\n  return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as
if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentHashCode(): Int {\n  return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentHashCode(): Int {\n  return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray?.contentHashCode(): Int {\n  return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a
hash code based on the contents of this array as if it is [List].\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray?.contentHashCode(): Int {\n  return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to

```



```

avoid deprecation warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentToString(): String {\n    return
    this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentToString(): String {\n    return
    this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentToString(): String {\n    return
    this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentToString(): String {\n    return
    this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the
contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray?.contentToString(): String
{\n    return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the
contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Copies this array or its subrange into the
[destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even
specify the subrange so that it overlaps with
the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position
in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to
copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n
*/\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out
of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is
out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
size): UIntArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return
destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n *
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the

```

[destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified

[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):  
ULongArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return  
destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the  
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the  
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or  
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex  
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array  
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):  
UByteArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return  
destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even  
specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy  
to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param  
startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive)  
of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or  
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex  
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array  
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UShortArray.copyInto(destination: UShortArray, destinationOffset:  
Int = 0, startIndex: Int = 0, endIndex: Int = size): UShortArray {\n    storage.copyInto(destination.storage,  
destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Returns new array which is a copy of  
the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UIntArray.copyOf(): UIntArray {\n    return UIntArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is  
a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
ULongArray.copyOf(): ULongArray {\n    return ULongArray(storage.copyOf())\n}\n\n/**\n * Returns new array  
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic  
inline fun UByteArray.copyOf(): UByteArray {\n    return UByteArray(storage.copyOf())\n}\n\n/**\n * Returns  
new array which is a copy of the original array.\n * \n * @sample  
samples.collections.Arrays.CopyOfOperations.copyOf\n
```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyOfOf(): UShortArray {\n return UShortArray(storage.copyOfOf())\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyOfOf(newSize: Int): UIntArray {\n return UIntArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyOfOf(newSize: Int): ULongArray {\n return ULongArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyOfOf(newSize: Int): UByteArray {\n return UByteArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyOfOf(newSize: Int, fromIndex: Int, toIndex: Int): UShortArray {\n return UShortArray(storage.copyOfOf(newSize, fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * - @param fromIndex the start of the range (inclusive) to copy.\n * - @param toIndex the end of the range (exclusive) to copy.\n * - @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * - @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyOfOfRange(fromIndex: Int, toIndex: Int): UIntArray {\n return UIntArray(storage.copyOfOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * - @param fromIndex the start of the range (inclusive) to copy.\n * - @param toIndex the end of the range (exclusive) to copy.\n * - @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * - @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyOfOfRange(fromIndex: Int, toIndex: Int): ULongArray {\n return ULongArray(storage.copyOfOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * - @param fromIndex the start of the range (inclusive) to copy.\n * - @param toIndex the end of the range (exclusive) to copy.\n * - @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * - @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyOfOfRange(fromIndex: Int, toIndex: Int): UByteArray {\n return

```

UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
 * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the
start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {\n    return
UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the
specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n *
@param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.fill(element:
UInt, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toInt(), fromIndex,
toIndex)\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex
the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.fill(element:
ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toLong(), fromIndex,
toIndex)\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex
the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.fill(element: UByte,
fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toByte(), fromIndex, toIndex)\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex]
is greater than [toIndex].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toShort(),
fromIndex, toIndex)\n}\n\n/**\n * Returns the range of valid indices for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.indices: IntRange\n    get()
= storage.indices\n}\n\n/**\n * Returns the range of valid indices for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.indices: IntRange\n    get()
= storage.indices\n}\n\n/**\n * Returns the range of valid indices for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UByteArray.indices: IntRange\n    get()
= storage.indices\n}\n\n/**\n * Returns the last valid index for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.lastIndex: Int\n    get() =
storage.lastIndex\n}\n\n/**\n * Returns the last valid index for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.lastIndex: Int\n    get() =
storage.lastIndex\n}\n\n/**\n * Returns the last valid index for the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UByteArray.lastIndex: Int\n    get() =
storage.lastIndex\n}\n\n/**\n * Returns the last valid index for the array.\n

```

```

*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UShortArray.lastIndex: Int\n  get() =
storage.lastIndex\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline operator fun UIntArray.plus(element: UInt): UIntArray {\n  return UIntArray(storage +
element.toInt())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun ULongArray.plus(element: ULong): ULongArray {\n  return ULongArray(storage +
element.toLong())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun UByteArray.plus(element: UByte): UByteArray {\n  return UByteArray(storage +
element.toByte())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline operator fun UShortArray.plus(element: UShort): UShortArray {\n  return UShortArray(storage +
element.toShort())\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator
fun UIntArray.plus(elements: Collection<UInt>): UIntArray {\n  var index = size\n  val result =
storage.copyOfOf(size + elements.size)\n  for (element in elements) result[index++] = element.toInt()\n  return
UIntArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator
fun ULongArray.plus(elements: Collection<ULong>): ULongArray {\n  var index = size\n  val result =
storage.copyOfOf(size + elements.size)\n  for (element in elements) result[index++] = element.toLong()\n
return ULongArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then
all elements of the given [elements] collection.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator fun UByteArray.plus(elements:
Collection<UByte>): UByteArray {\n  var index = size\n  val result = storage.copyOfOf(size + elements.size)\n
for (element in elements) result[index++] = element.toByte()\n  return UByteArray(result)\n}\n\n/**\n * Returns
an array containing all elements of the original array and then all elements of the given [elements] collection.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator fun UShortArray.plus(elements:
Collection<UShort>): UShortArray {\n  var index = size\n  val result = storage.copyOfOf(size + elements.size)\n
for (element in elements) result[index++] = element.toShort()\n  return UShortArray(result)\n}\n\n/**\n * Returns
an array containing all elements of the original array
and then all elements of the given [elements] array.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UIntArray.plus(elements: UIntArray): UIntArray {\n  return UIntArray(storage + elements.storage)\n}\n\n/**\n *
Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
ULongArray.plus(elements: ULongArray): ULongArray {\n  return ULongArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] array.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UByteArray.plus(elements: UByteArray): UByteArray {\n  return UByteArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] array.\n\n*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UShortArray.plus(elements: UShortArray): UShortArray {\n  return UShortArray(storage +
elements.storage)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic
fun UIntArray.sort(): Unit {\n  if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n *

```

```

@sample samples.collections.Arrays.Sorting.sortArray\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(): Unit {\n  if (size > 1)
sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UByteArray.sort(): Unit {\n  if (size >
1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.sort(): Unit {\n  if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts a range in the array in-
place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end
of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(fromIndex: Int = 0, toIndex:
Int = size): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArray(this, fromIndex,
toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArray(this, fromIndex, toIndex)\n}\n\n/**\n *
Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their
natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end
of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n  sort(fromIndex, toIndex)\n  reverse(fromIndex,
toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements
are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range
(inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *

```

@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.toByteArray(): ByteArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.toIntArray(): IntArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.toLongArray(): LongArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.toShortArray(): ShortArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.toTypedArray(): Array<ULong> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray(): Array<UByte> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray(): Array<UShort> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of UByte containing all of the elements of this generic array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UByte>.toUByteArray(): UByteArray {\n    return UByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.toUByteArray(): UByteArray {\n
```

```

return UByteArray(this.copyOf())\n\n/**\n * Returns an array of UInt containing all of the elements of this
generic array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out
UInt>.toUIntArray(): UIntArray {\n    return UIntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array
of type [UIntArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.toUIntArray(): UIntArray {\n    return UIntArray(this.copyOf())\n}\n\n/**\n * Returns an array of ULong
containing all of the elements of this generic array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out ULong>.toULongArray():
ULongArray {\n    return ULongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type
[ULongArray], which is a copy of this array where
each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.toULongArray(): ULongArray {\n    return ULongArray(this.copyOf())\n}\n\n/**\n * Returns an array
of UShort containing all of the elements of this generic array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UShort>.toUShortArray():
UShortArray {\n    return UShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type
[UShortArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.toUShortArray(): UShortArray {\n    return UShortArray(this.copyOf())\n}\n\n/**\n * Returns a [Map]
where keys are elements from the given array and values
are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the
last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n *
*/\n@sample samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {\n    val result = LinkedHashMap<UInt,
V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a
[Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function
applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {\n    val result =
LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {\n    val result =
LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n    val result =

```



```

LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n  return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any
two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n for
(element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination}\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n *
where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If
any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <V, M : MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong)
-> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return
destination}\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n
for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return
destination}\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is
the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two
elements are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M
{\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return
destination}\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {\n return
flatMapTo(ArrayList<R>(), transform)\n}\n}\n\n/**\n * Returns a single list of all elements yielded from results of
[transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n
return flatMapTo(ArrayList<R>(), transform)\n}\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element of original array.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMapIndexed(transform: (index: Int, UInt)
-> Iterable<R>): List<R> {\n  return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a
single list of all elements yielded from results of [transform] function being invoked on each element\n * and its
index in the original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its
index in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) ->
Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++, element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapIndexedTo(destination:
C, transform: (index: Int, ULong) -> Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list =
transform(index++, element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element\n * and its index in the original
array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, UByte) ->
Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++, element)\n

```

```

destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each
element\n * and its index in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) ->
Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {\n for
(element in this) {\n val list =
transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements
yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) ->
Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given
[keySelector] function\n * applied to each element and returns a map where each group key is associated with a list
of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from
the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {\n return groupByTo(LinkedHashMap<K,
MutableList<UInt>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {\n return
groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {\n return
groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map

```

where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample

```

samples.collections.Collections.Transformations.groupBy\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <K> UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {\n return
groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)\n}\n\n/**\n * Groups values returned by
the [valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of
corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n
* Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the
key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key
is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the
keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n
* Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the
key returned by the given [keySelector] function applied to the element\n * and returns
a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves
the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n
* Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the
key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key
is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the
keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n
* Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {\n
for (element in this) {\n val key = keySelector(element)\n val
list = destination.getOrPut(key) { ArrayList<UInt>() }\n list.add(element)\n }\n return
destination\n}\n\n/**\n
* Groups elements of the original array by the key returned by the given [keySelector]
function\n * applied to each element and puts to the [destination] map each group key associated with a list of
corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n

```

```

*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <K, M :
MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K):
M {n for (element in this) {n val key = keySelector(element)n val list = destination.getOrPut(key) {
ArrayList<ULong>() }n list.add(element)n }n return destinationn}n\/n\/**n * Groups elements of the
original array by the key returned by the given [keySelector] functionn * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.n * n *
@return The [destination] map.n * n * @sample samples.collections.Collections.Transformations.groupByn
*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <K, M :
MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K):
M {n for (element in this) {n val key = keySelector(element)n val list = destination.getOrPut(key) {
ArrayList<UByte>() }n list.add(element)n }n return destinationn}n\/n\/**n * Groups elements of the
original array by the key returned by the given [keySelector] functionn * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.n * n * @return The
[destination] map.n * n * @sample samples.collections.Collections.Transformations.groupByn
*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic
inline fun <K, M : MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M,
keySelector: (UShort) -> K): M {n for (element in this) {n val key = keySelector(element)n val list =
destination.getOrPut(key) { ArrayList<UShort>() }n list.add(element)n }n return destinationn}n\/n\/**n
* Groups values returned by the [valueTransform] function applied to each element of the original arrayn * by the
key returned by the given [keySelector] function applied to the elementn * and puts to the [destination] map each
group key associated with a list of corresponding values.n * n * @return The [destination] map.n * n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValuesn
*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UIntArray.groupByTo(destination:
M, keySelector: (UInt) -> K, valueTransform: (UInt) -> V): M {n for (element in this) {n val key =
keySelector(element)n val list = destination.getOrPut(key) { ArrayList<V>() }n
list.add(valueTransform(element))n }n return destinationn}n\/n\/**n * Groups values returned by the
[valueTransform] function applied to each element of the original arrayn * by the key returned by the given
[keySelector] function applied to the elementn * and puts to the [destination] map each group key associated with a
list of corresponding values.n * n * @return The [destination] map.n * n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValuesn
*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K,
valueTransform: (ULong) -> V): M {n for (element in this) {n val
key = keySelector(element)n val list = destination.getOrPut(key) { ArrayList<V>() }n
list.add(valueTransform(element))n }n return destinationn}n\/n\/**n * Groups values returned by the
[valueTransform] function applied to each element of the original arrayn * by the key returned by the given
[keySelector] function applied to the elementn * and puts to the [destination] map each group key associated with a
list of corresponding values.n * n * @return The [destination] map.n * n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValuesn
*\/n@SinceKotlin("1.3\/")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K,
valueTransform: (UByte) -> V): M {n for (element in this) {n val key = keySelector(element)n val list
= destination.getOrPut(key) { ArrayList<V>() }n list.add(valueTransform(element))n
}n return destinationn}n\/n\/**n * Groups values returned by the [valueTransform] function applied to each
element of the original arrayn * by the key returned by the given [keySelector] function applied to the elementn *
and puts to the [destination] map each group key associated with a list of corresponding values.n * n * @return
The [destination] map.n * n * @sample

```

```

samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UShortArray.groupByKeyTo(destination: M, keySelector: (UShort) -> K,
valueTransform: (UShort) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list
= destination.getOrPut(key) { ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return
destination}\n}\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.map(transform: (UInt) -> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to each element in the original
array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.map(transform: (ULong) -> R): List<R> {\n    return mapTo(ArrayList<R>(size),
transform)\n}\n}\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UByteArray.map(transform: (UByte) -> R): List<R> {\n    return mapTo(ArrayList<R>(size),
transform)\n}\n}\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.map(transform: (UShort) -> R): List<R> {\n    return mapTo(ArrayList<R>(size),
transform)\n}\n}\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element and its index in the original array.\n * \n * @param [transform] function that takes the index of an element and
the element itself\n * and returns the result of the transform applied to the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UIntArray.mapIndexed(transform: (index: Int, UInt) -> R): List<R> {\n    return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * \n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> ULongArray.mapIndexed(transform: (index: Int, ULong) -> R): List<R> {\n    return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * \n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UByteArray.mapIndexed(transform: (index: Int, UByte) -> R): List<R> {\n    return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * \n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UShortArray.mapIndexed(transform: (index: Int, UShort) -> R): List<R> {\n    return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Applies the given [transform] function to each
element and its index in the original array\n * and appends the results to the given [destination].\n * \n * @param
[transform] function that takes the index of an element and
the element itself\n * and returns the result of the transform applied to the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {\n    var

```

```

index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.withIndex(): Iterable<IndexedValue<UInt>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */

```

```

*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic fun UByteArray.withIndex():
Iterable<IndexedValue<UByte>> {n return IndexingIterable { iterator() }n}n/n/**n * Returns a lazy [Iterable]
that wraps each element of the original arrayn * into an [IndexedValue] containing the index of that element and the
element itself.n */n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic fun UShortArray.withIndex():
Iterable<IndexedValue<UShort>> {n return IndexingIterable { iterator() }n}n/n/**n * Returns `true` if all
elements match the given [predicate].n * n * @sample samples.collections.Collections.Aggregates.alln
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UIntArray.all(predicate: (UInt) -> Boolean): Boolean {n for (element in this) if (!predicate(element)) return
false\n return true\n}n/n/**n * Returns `true` if all elements match the given [predicate].n * n * @sample
samples.collections.Collections.Aggregates.alln
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
ULongArray.all(predicate: (ULong) -> Boolean): Boolean {n for (element in this) if (!predicate(element)) return
false\n return true\n}n/n/**n * Returns `true` if all elements match the given [predicate].n * n * @sample
samples.collections.Collections.Aggregates.alln
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {n for (element in this) if (!predicate(element)) return
false\n return true\n}n/n/**n * Returns `true` if all elements match the given [predicate].n * n * @sample
samples.collections.Collections.Aggregates.alln
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {n for (element in this) if (!predicate(element))
return false\n return true\n}n/n/**n * Returns `true` if array has at least one element.n * n * @sample
samples.collections.Collections.Aggregates.any\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UIntArray.any(): Boolean {n return storage.any()\n}n/n/**n * Returns `true` if array has at least one element.n
* n * @sample samples.collections.Collections.Aggregates.any\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
ULongArray.any(): Boolean {n return storage.any()\n}n/n/**n * Returns `true` if array has at least one
element.n * n * @sample samples.collections.Collections.Aggregates.any\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UByteArray.any(): Boolean {n return storage.any()\n}n/n/**n * Returns `true` if array has at least one
element.n * n * @sample samples.collections.Collections.Aggregates.any\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UShortArray.any(): Boolean {n return storage.any()\n}n/n/**n * Returns `true` if at least one element matches
the given [predicate].n * n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UIntArray.any(predicate: (UInt) -> Boolean): Boolean {n for (element in this) if (predicate(element)) return true\n
return false\n}n/n/**n * Returns `true` if at least one element matches the given [predicate].n * n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
ULongArray.any(predicate: (ULong) -> Boolean): Boolean {n for (element in this) if (predicate(element)) return
true\n return false\n}n/n/**n * Returns
`true` if at least one element matches the given [predicate].n * n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UByteArray.any(predicate: (UByte) -> Boolean): Boolean {n for (element in this) if (predicate(element)) return
true\n return false\n}n/n/**n * Returns `true` if at least one element matches the given [predicate].n * n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n
*/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun

```



```

UShortArray.any(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

Returns the number of elements matching the given [predicate].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UIntArray.count(predicate: (UInt)
-> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

Returns the number of elements matching the given [predicate].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ULongArray.count(predicate: (ULong) -> Boolean): Int {
    var count = 0
    for (element in this) if
(predicate(element)) ++count
    return count
}

Returns the number of elements matching the given
[predicate].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public
inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if
(predicate(element)) ++count
    return count
}

Returns the number of elements matching the given
[predicate].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public
inline fun UShortArray.count(predicate: (UShort)
-> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return
count
}

Accumulates value starting with [initial] value and applying [operation] from left to right
to current accumulator value and each element.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this)
accumulator = operation(accumulator, element)
    return accumulator
}

Accumulates value starting
with [initial] value and applying [operation] from left to right
to current accumulator value and each element.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in
this) accumulator = operation(accumulator, element)
    return accumulator
}

Accumulates value
starting with [initial] value and applying [operation] from left to right
to current accumulator value and each
element.

Returns the specified [initial] value if the array is empty.

@param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this)
accumulator = operation(accumulator, element)
    return accumulator
}

Accumulates value starting with [initial] value and applying [operation] from left to right
to current accumulator
value and each element.

Returns the specified [initial] value if the array is empty.

@param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
<R> UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {
    var accumulator = initial
    for
(element in this) accumulator = operation(accumulator, element)
    return accumulator
}

Accumulates
value starting with [initial] value and applying [operation] from left to right
to current accumulator value and
each element with its index in the original array.

Returns the specified [initial]
value if the array is empty.

@param [operation] function that takes the index of an element, current
accumulator value
and the element itself, and calculates the next accumulator value.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = 0
    var
accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return
}

```

```

accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic
inline fun <R> ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {\n  var index
= 0\n  var accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n
return accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left
to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic inline fun <R>
UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {\n  var index = 0\n  var
accumulator = initial\n
for (element in this) accumulator = operation(index++, accumulator, element)\n  return accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current
accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value
if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator
value\n * and the element itself, and calculates the next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic inline fun <R>
UShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): R {\n  var index = 0\n  var
accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic inline fun <R>
UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator =
initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic inline fun <R>
ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n
return accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right
to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@\n@SinceKotlin("1.3")\n@\n@ExperimentalUnsignedTypes\n@\n@kotlin.internal.InlineOnly\n\npublic inline fun <R>
UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n
}\n  return accumulator\n}\n\n\n/**\n * Accumulates value starting with [initial] value and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and
calculates the next accumulator value.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right
to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the
specified [initial] value if the
array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and
current accumulator value, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {\n  var index = lastIndex\n
var accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n
--index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n
*\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator
value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {\n  var index =
lastIndex\n  var accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from right to left\n * to each element with its index in the original array and current
accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.foldRightIndexed(initial: R, operation:
(index: Int, UByte, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator = initial\n  while (index >= 0)
{\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {\n  var index =
lastIndex\n  var accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Performs the given [action] on each
element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.forEach(action: (UInt) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.forEach(action: (ULong) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.forEach(action: (UByte) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UShortArray.forEach(action: (UShort) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element, providing sequential index with the element.\n * @param [action]
function that takes the index of an element and the element itself\n * and performs the action on the element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit): Unit {\n  var index = 0\n  for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {\n  var index
= 0\n  for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element,
providing sequential index with the element.\n * @param [action] function that takes the index of an element and
the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {\n  var index = 0\n  for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {\n  var index = 0\n  for (item in this)
action(index++, item)\n}\n\n@Deprecated("Use
maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.max(): UInt? {\n  return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.max():
ULong? {\n  return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.max():
UByte? {\n  return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.max(): UShort? {\n  return maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt? {\n  return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong? {\n  return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince
= "1.4", errorSince = "1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {\n  return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

<R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {\n  return
maxByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {\n  if (isEmpty()) return
null\n  var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var
maxValue = selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if
(maxValue < v) {\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {\n  if (isEmpty()) return null\n
var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v) {\n      maxElem
= e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the
largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {\n  if (isEmpty()) return null\n
var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {\n  if (isEmpty()) return null\n
var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if
the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOf(selector: (UInt) -> Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n
var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue =
maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.maxOf(selector: (ULong) -> Double): Double {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the
largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws

```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.maxOf(selector: (UByte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by\n * [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is\n * empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOf(selector: (UShort) -> Double): Double {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the\n * largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If\n * any of values produced by [selector]
```

```
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.maxOf(selector: (UInt) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n * function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun\nULongArray.maxOf(selector: (ULong) -> Float): Float {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the\n * largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If\n * any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.maxOf(selector: (UByte) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return\n    maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to\n * each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n * `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOf(selector: (UShort) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n * by [selector]
```

```
function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is\n * empty.\n */
```

```

*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun <R :
Comparable<R>> UIntArray.maxOf(selector: (UInt) -> R): R {^n if (isEmpty()) throw
NoSuchElementException()^n var maxValue = selector(this[0])^n for (i in 1..lastIndex) {^n val v =
selector(this[i])^n if (maxValue < v) {^n maxValue = v^n }^n }^n return maxValue^n}^n/**^n *
Returns the largest value among all values produced by [selector] function^n * applied to each element in the
array.^n * ^n * @throws NoSuchElementException if the array is empty.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public
inline fun <R : Comparable<R>> ULongArray.maxOf(selector: (ULong) -> R): R {^n if (isEmpty()) throw
NoSuchElementException()^n var maxValue = selector(this[0])^n for (i in 1..lastIndex) {^n val v =
selector(this[i])^n if (maxValue < v) {^n maxValue = v^n }^n }^n return maxValue^n}^n/**^n *
Returns the largest value among all values produced by [selector] function^n * applied to each element in the
array.^n * ^n * @throws NoSuchElementException if the array is empty.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun <R :
Comparable<R>> UByteArray.maxOf(selector: (UByte) -> R): R {^n if (isEmpty()) throw
NoSuchElementException()^n var maxValue = selector(this[0])^n for (i in 1..lastIndex) {^n val v =
selector(this[i])^n if (maxValue < v) {^n
maxValue = v^n }^n }^n return maxValue^n}^n/**^n * Returns the largest value among all values
produced by [selector] function^n * applied to each element in the array.^n * ^n * @throws
NoSuchElementException if the array is empty.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun <R :
Comparable<R>> UShortArray.maxOf(selector: (UShort) -> R): R {^n if (isEmpty()) throw
NoSuchElementException()^n var maxValue = selector(this[0])^n for (i in 1..lastIndex) {^n val v =
selector(this[i])^n if (maxValue < v) {^n maxValue = v^n }^n }^n return maxValue^n}^n/**^n *
Returns the largest value among all values produced by [selector] function^n * applied to each element in the array
or `null` if there are no elements.^n * ^n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {^n if (isEmpty()) return null^n var maxValue =
selector(this[0])^n for (i in 1..lastIndex) {^n val v = selector(this[i])^n maxValue = maxOf(maxValue, v)^n
}^n return maxValue^n}^n/**^n * Returns the largest value among all values produced by [selector] function^n *
applied to each element in the array or `null` if there are no elements.^n * ^n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
ULongArray.maxOfOrNull(selector:
(ULong) -> Double): Double? {^n if (isEmpty()) return null^n var maxValue = selector(this[0])^n for (i in
1..lastIndex) {^n val v = selector(this[i])^n maxValue = maxOf(maxValue, v)^n }^n return
maxValue^n}^n/**^n * Returns the largest value among all values produced by [selector] function^n * applied to
each element in the array or `null` if there are no elements.^n * ^n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.^n
*^@SinceKotlin("1.4")^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)^@OverloadResolution
ByLambdaReturnType^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {^n if (isEmpty()) return null^n var maxValue

```

```
= selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
```

the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`\n if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of\n values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun\nUIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of\n values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {\n    if (isEmpty()) return\n    null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue\n        = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values\n produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If\n any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by\n [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOfOrNull(selector: (UShort) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun <R : Comparable<R>> UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {\n    if (isEmpty()) return\n    null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value\n among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no\n elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
```



```

Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {\n  if (isEmpty()) return null\n  var\n  maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n*\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :\nComparable<R>> UByteArray.maxOfOrNull(selector: (UByte) -> R): R? {\n  if (isEmpty()) return null\n  var\n  maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n*\n * Returns the largest value among all values\n produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun <R : Comparable<R>> UShortArray.maxOfOrNull(selector: (UShort) -> R): R? {\n  if (isEmpty())\n  return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if\n (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n*\n * Returns the largest value\n according to the provided [comparator]\n * among all values produced by [selector] function applied to each\n element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n  if (isEmpty()) throw\n  NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v\n = selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return\n  maxValue\n}\n\n*\n * Returns the largest value according to the provided [comparator]\n * among all\n values produced by [selector] function applied to each element in the array.\n * \n * @throws\n NoSuchElementException if the array is empty.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n  if (isEmpty()) throw\n  NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =\n selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return\n  maxValue\n}\n\n*\n * Returns the largest value according to the provided [comparator]\n * among all\n values produced by [selector] function applied to each element in the array.\n * \n * @throws\n NoSuchElementException if the array is empty.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n  if (isEmpty()) throw\n  NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =\n selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return\n  maxValue\n}\n\n*\n * Returns the largest value according to the provided [comparator]\n * among all\n values produced by [selector] function applied to each element in the array.\n * \n * @throws\n NoSuchElementException if\n the array is empty.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun <R> UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n  if\n (isEmpty()) throw\n  NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return\n  maxValue\n}\n\n*\n * Returns the largest value according to the provided [comparator]\n * among all

```

```

values produced by [selector] function applied to each element in the array or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector:
(UInt) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n
    val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n
return maxValue\n}\n\n**\n * Returns the largest value according to the provided [comparator]\n * among all
values produced by [selector] function applied to each element in the array or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n**\n * Returns the largest value according to the provided [comparator]\n * among
all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n
if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n**\n * Returns the largest element or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxOrNull(): UInt? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max
= e\n    }\n    return max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.maxOrNull(): ULong? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in
1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n**\n * Returns the largest
element or `null` if there are no elements.
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.maxOrNull(): UByte? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n
        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n**\n * Returns the largest element or `null`
if there are no elements.
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.maxOrNull(): UShort? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex)
{\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n@Deprecated("Use maxWithOrNull
instead.", ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince
= "1.4", errorSince = "1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWith(comparator:
Comparator<in UInt>): UInt? {\n    return maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull
instead.", ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.maxWith(comparator: Comparator<in ULong>): ULong? {\n    return
maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.",

```

```

ReplaceWith("this.maxOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.maxWith(comparator: Comparator<in UByte>): UByte? {\n    return
maxOrNull(comparator)\n}\n\n@Deprecated("Use maxOrNull
instead.", ReplaceWith("this.maxOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.maxWith(comparator: Comparator<in UShort>): UShort? {\n    return
maxOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWithOrNull(comparator:
Comparator<in UInt>): UInt? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n
val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.maxWithOrNull(comparator: Comparator<in ULong>): ULong? {\n    if (isEmpty()) return null\n
var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max =
e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided
[comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n
val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxWithOrNull(comparator:
Comparator<in UShort>): UShort?
{\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.min(): UInt?
{\n    return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.min():
ULong? {\n    return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.min(): UByte? {\n    return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.min():
UShort? {\n    return minOrNull()\n}\n\n@Deprecated("Use minByOrNull instead.",
ReplaceWith("this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UIntArray.minBy(selector: (UInt) -> R): UInt? {\n    return
minByOrNull(selector)\n}\n\n@Deprecated("Use minByOrNull instead.",
ReplaceWith("this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> ULongArray.minBy(selector: (ULong) -> R): ULong? {\n    return
minByOrNull(selector)\n}\n\n@Deprecated("Use minByOrNull instead.",

```

```

ReplaceWith("\this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte? {\n  return
minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith("\this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort? {\n  return
minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minByOrNull(selector: (UInt) -> R): UInt? {\n  if (isEmpty()) return null\n  var
minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minByOrNull(selector:
(ULong) -> R): ULong? {\n  if (isEmpty()) return null\n  var minElem = this[0]\n  val lastIndex =
this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue = selector(minElem)\n  for (i in
1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v) {\n      minElem = e\n
minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value
of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minByOrNull(selector: (UByte) -> R): UByte? {\n  if (isEmpty()) return null\n
var minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function
or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.minByOrNull(selector: (UShort) -> R): UShort? {\n  if (isEmpty()) return null\n
var minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector]
function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`,
the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.minOf(selector: (UInt) -> Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue =
minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]

```

```

function is `NaN`, the returned result is `NaN`. \n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.minOf(selector: (ULong) -> Double): Double {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`. \n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOf(selector: (UByte)
-> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n
for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`. \n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOf(selector: (UShort) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`,
the returned result is `NaN`. \n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.minOf(selector: (UInt) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the
returned result is `NaN`. \n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOf(selector: (ULong) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`. \n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.minOf(selector: (UByte) -> Float): Float {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`. \n * \n * @throws
NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOf(selector: (UShort) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i
in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minOf(selector: (UByte)
-> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n
return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or
`null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun ULongArray.minOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.minOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty())
        return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.minOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.minOrNull(selector: (UInt) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.minOrNull(selector: (ULong) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.minOrNull(selector: (UByte) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.minOrNull(selector: (UShort) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

```

* Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

if there are no elements.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minOfOrNull(selector: (UInt) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minOfOrNull(selector: (ULong) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value\n * among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no\n * elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minOfOrNull(selector: (UByte) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element\n * in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minOfOrNull(selector: (UShort) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the\n * provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n<R> UIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among\n * all values produced by [selector] function applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
```



```

selector(this[i])\n    if (comparator.compare(minValue, v) > 0) {\n        minValue = v\n    }\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n
*\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n    if
(isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n    if
(isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minOrNull(): UInt? {\n    if
(isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n return min\n}\n\n/**\n * Returns the
smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minOrNull(): ULong? {\n    if
(isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min =
e\n    }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minOrNull(): UByte? {\n    if
(isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min =
e\n    }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minOrNull(): UShort? {\n if
(isEmpty()) return
null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return
min\n}\n\n@Deprecated("Use minWithOrNull instead."),
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.minWith(comparator: Comparator<in UInt>): UInt? {\n return
minWithOrNull(comparator)\n}\n\n@Deprecated("Use minWithOrNull instead."),
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.minWith(comparator: Comparator<in ULong>): ULong? {\n return
minWithOrNull(comparator)\n}\n\n@Deprecated("Use minWithOrNull instead."),
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.minWith(comparator: Comparator<in UByte>): UByte? {\n return
minWithOrNull(comparator)\n}\n\n@Deprecated("Use minWithOrNull instead."),
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.minWith(comparator: Comparator<in UShort>): UShort? {\n return
minWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the smallest value according to the
provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minWithOrNull(comparator:
Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n
}\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minWithOrNull(comparator:
Comparator<in ULong>): ULong? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the
first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the
first element having
the smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minWithOrNull(comparator:
Comparator<in UShort>): UShort? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns
`true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns
`true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.none(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n
* \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.none(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.none(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(predicate:
(UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return false\n    return
true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.onEach(action: (UInt) -> Unit): UIntArray {\n    return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.onEach(action: (ULong) -> Unit): ULongArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray
{\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each
element and returns the array itself afterwards.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.onEach(action: (UShort) -> Unit): UShortArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the
element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element
and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index

```

with the

element, and returns the array itself afterwards. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {\n    return apply {\n        forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]  
from left to right to current accumulator value and each element.\n * \n * Throws an exception if this array is  
empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its  
receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,  
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic  
inline fun UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,  
and calculates the next  
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic  
inline fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for  
(index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return  
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to  
right to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the  
array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is  
empty.\n * \n * @param [operation] function that takes current accumulator value and an element,  
and calculates  
the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline  
fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right to current  
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty  
in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,  
and calculates the next  
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UShortArray.reduce(operation:  
(acc: UShort, UShort) -> UShort): UShort {\n    if (isEmpty())\n        throw  
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in  
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right to current  
accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is  
empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null`  
when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current  
accumulator value and the element itself,  
and calculates the next accumulator value.\n * \n * @sample  
samples.collections.Collections.Aggregates.reduce\n */
```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.reduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt {\n  if (isEmpty())\n  throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = this[0]\n  for
(index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return
accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong {\n  if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = this[0]\n  for
(index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return
accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value and the element itself,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte {\n  if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = this[0]\n  for
(index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return
accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an
element,
current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort {\n  if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = this[0]\n  for
(index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return
accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null`
if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator
value and the element itself,\n * and
calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt? {\n  if (isEmpty())\n
return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index,
accumulator, this[index])\n  }\n  return accumulator}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element with its index
in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator

```

```

value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {\n
if (isEmpty())\n    return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator =
operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element
with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceIndexedOrNull(operation:
(index: Int, acc: UByte, UByte) -> UByte): UByte? {\n    if (isEmpty())\n        return null\n    var accumulator =
this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from
left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns
`null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceIndexedOrNull(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort? {\n    if
(isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index,
accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt?
{\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n
accumulator = operation(accumulator, this[index])\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) ->
ULong): ULong? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex)
{\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates
value starting with the first element and applying [operation] from left to right\n * to current accumulator value and
each
element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) ->
UByte): UByte? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex)
{\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates

```

value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @WasExperimental(ExperimentalStdlibApi::class)\n * @kotlin.internal.InlineOnly\n * public inline fun UShortArray.reduceOrNull(operation: (acc: UShort, UShort) -> UShort): UShort? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex)\n * {\n * accumulator = operation(accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {\n * var index = lastIndex\n * if (index < 0)\n * throw UnsupportedOperationException("Empty array can't be reduced.")\n * var accumulator = get(index--)\n * while (index >= 0)\n * {\n * accumulator = operation(get(index--), accumulator)\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {\n * var index = lastIndex\n * if (index < 0)\n * throw UnsupportedOperationException("Empty array can't be reduced.")\n * var accumulator = get(index--)\n * while (index >= 0)\n * {\n * accumulator = operation(get(index--), accumulator)\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {\n * var index = lastIndex\n * if (index < 0)\n * throw UnsupportedOperationException("Empty array can't be reduced.")\n * var accumulator = get(index--)\n * while (index >= 0)\n * {\n * accumulator = operation(get(index--), accumulator)\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {\n * var index = lastIndex\n * if (index < 0)\n * throw UnsupportedOperationException("Empty array can't be reduced.")\n * var accumulator = get(index--)\n * while (index >= 0)\n * {\n * accumulator = operation(get(index--), accumulator)\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to

left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use `reduceRightIndexedOrNull` instead. It returns `null` when its receiver is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.reduceRight
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator =
        get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
*/
*/ Accumulates value starting with the last element and applying [operation]
from right to left to each element with its index in the original array and current accumulator value.
Throws an exception if this array is empty. If the array can be empty in an expected way, please use
[reduceRightIndexedOrNull]
instead. It returns `null` when its receiver is empty. @param [operation] function that takes the index of an
element, the element itself and current accumulator value, and calculates the next accumulator value.
@sample samples.collections.Collections.Aggregates.reduceRight
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {
    var index =
        lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
        accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
            accumulator)
        --index
    }
    return accumulator
}
*/
*/ Accumulates value starting with the last
element and applying [operation] from right to left to each element with its index in the original array and
current accumulator value.
Throws
an exception if this array is empty. If the array can be empty in an expected way, please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty. @param [operation]
function that takes the index of an element, the element itself and current accumulator value, and calculates the
next accumulator value.
@sample samples.collections.Collections.Aggregates.reduceRight
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {
    var index =
        lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
        accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
            accumulator)
        --index
    }
    return accumulator
}
*/
*/ Accumulates value starting with the last
element and applying [operation]
from right to left to each element with its index in the original array and current accumulator value.
Throws an exception if this array is empty. If the array can be empty in an expected way, please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty. @param [operation]
function that takes the index of an element, the element itself and current accumulator value, and calculates the
next accumulator value.
@sample samples.collections.Collections.Aggregates.reduceRight
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {
    var index =
        lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
        accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
            accumulator)
        --index
    }
    return accumulator
}
*/
*/ Accumulates value starting with the last element and applying
[operation] from right to left to each element with its index in the original array and current accumulator value.
*/
*/ Returns `null` if the array is empty. @param [operation] function that takes the index of an element,
the element itself and current accumulator value, and calculates the next accumulator value.
@sample

```


samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {\n    var index =  
lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting
```

with the last element and applying [operation] from right to left\n * to each element with its index in the original
array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {\n    var  
index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n *  
* to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the  
array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and  
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull

samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {\n    var  
index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with  
its index in the original array and current accumulator
```

value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UShortArray.reduceRightIndexedOrNull(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort? {\n    var  
index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and  
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that  
takes
```

an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt):  
UInt? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index  
>= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and  
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that  
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull

samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) ->  
ULong): ULong? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
```


element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUShortArray.runningFold(initial:
```

```
R, operation: (acc: R, UShort) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n
```

```
\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note
```

```
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.runningFold

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {\n    if\n    (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator =\n    initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    result.add(accumulator)\n}\nreturn result\n}\n\n/**\n * Returns a list containing successive accumulation\n    values generated by applying [operation] from left to
```

```
right\n * to each element and current accumulator value that starts with the first element of this array.\n * Note\n    that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous\n    value in resulting list.\n * @param [operation] function that takes current accumulator value and an element,\n    and calculates the next accumulator value.\n * @sample
```

samples.collections.Collections.Aggregates.runningReduce

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {\n    if (isEmpty()) return\n    emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<UInt>(size).apply { add(accumulator) }\n    for\n    (index in 1 until size) {\n        accumulator = operation(accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
```

```
values generated by applying [operation] from left to right\n * to each element and current accumulator value that\n    starts with the first element of this array.\n * Note that `acc` value passed to [operation] function should not be\n    mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that\n    takes current accumulator value and an element, and calculates the next accumulator value.\n * @sample
```

samples.collections.Collections.Aggregates.runningReduce

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {\n    if (isEmpty()) return\n    emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<ULong>(size).apply { add(accumulator) }\n    for\n    (index in 1 until size) {\n        accumulator = operation(accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * Note that `acc`\n    value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in\n    resulting list.\n * @param [operation] function that takes current accumulator value and an element, and\n    calculates the next accumulator value.\n * @sample
```

```
samples.collections.Collections.Aggregates.runningReduce\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * Note that `acc`\n    value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in\n    resulting list.\n * @param [operation] function that takes current accumulator value and an element, and\n    calculates the next accumulator value.\n * @sample
```

samples.collections.Collections.Aggregates.runningReduce

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {\n    if (isEmpty()) return\n    emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<UByte>(size).apply { add(accumulator) }\n    for\n    (index in 1 until size) {\n        accumulator = operation(accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
```

```
values generated by applying [operation] from left to right\n * to each element and current accumulator value that\n    starts with the first element of this array.\n * Note that `acc` value passed to [operation] function should not be\n    mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that\n    takes current accumulator value and an element, and calculates the next accumulator value.\n * @sample
```

samples.collections.Collections.Aggregates.runningReduce

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.runningReduce(operation: (acc: UShort, UShort) -> UShort): List<UShort> {\n    if (isEmpty()) return\n    emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<UShort>(size).apply { add(accumulator) }\n    for\n    (index in 1 until size)
```


function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scan(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
```

```

@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, ULong) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list
containing successive accumulation values generated
    by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator
value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value\n * and the element itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, UByte) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index
in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value
passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting
list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, UShort) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all
values produced by [selector] function applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf
instead.\n", ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun
UIntArray.sumBy(selector: (UInt) -> UInt): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf instead.\n",
ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun
ULongArray.sumBy(selector: (ULong) -> UInt): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf instead.\n",
ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun
UByteArray.sumBy(selector: (UByte) -> UInt): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf instead.\n",
ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun
UShortArray.sumBy(selector: (UShort) -> UInt): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf instead.\n",
ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun
UIntArray.sumByDouble(selector: (UInt) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n * \n * @Deprecated(\n"Use sumOf instead.\n",
ReplaceWith(\n"this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun

```

```

ULongArray.sumByDouble(selector: (ULong) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.sumByDouble(selector: (UByte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.sumByDouble(selector: (UShort) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.sumOf(selector: (UInt) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.sumOf(selector: (ULong) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.sumOf(selector: (UByte) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.sumOf(selector: (UShort) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.sumOf(selector: (UInt) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public

```



```

inline fun ULongArray.sumOf(selector: (ULong) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced by
[selector] function applied to each element in the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector:
(UByte) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced by [selector] function applied to each element in
the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public
inline fun UShortArray.sumOf(selector: (UShort) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced by
[selector] function applied to each element in the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfLong")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt)
-> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced
by [selector] function applied to each element in the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfLong")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector:
(ULong) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this)
        sum +=
selector(element)
    return sum
}
/** Returns the sum of all values produced by [selector] function
applied to each element in the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfLong")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector:
(UByte) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element
in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced
by [selector] function applied to each element in the array.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfLong")
@ExperimentalUnsignedTypes
kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector:
(UShort) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this)
        sum +=
selector(element)
    return sum
}
/** Returns the sum of all values produced by [selector] function
applied to each element in the array.

*/
@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfUInt")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
kotlin.internal.Inline
Only
public
inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this)
        sum += selector(element)
    return sum
}
/** Returns the sum of all values produced by
[selector] function applied to each element in the array.

*/
@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
kotlin.jvm.JvmName("sumOfUInt")

```

```

n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun ULongArray.sumOf(selector: (ULong) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n
for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all
values produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic
inline fun UByteArray.sumOf(selector: (UByte) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> ULong): ULong {\n    var sum: ULong =
0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns
the sum of all values produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic
inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {\n    var sum: ULong =
0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns
the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {\n    var sum: ULong =
0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns a
list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has
length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other: Array<out
R>): List<Pair<UInt, R>> {\n    return zip(other) { t1, t2 ->
t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the
same index.\n * The returned list has length of the shortest collection.\n * \n * @sample

```

```

samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Array<out R>): List<Pair<ULong, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has
length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other: Array<out
R>): List<Pair<UByte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other:
Array<out R>): List<Pair<UShort, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
values built from the elements of `this` array and the [other] array with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n
*\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
ULongArray.zip(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n
*\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n
*\n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of pairs built from the elements
of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n
*\n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other:
Iterable<R>): List<Pair<UInt, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Iterable<R>): List<Pair<ULong, R>> {\n

```



```

a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list
has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray.zip(other: UByteArray):
List<Pair<UByte, UByte>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray.zip(other: UShortArray):
List<Pair<UShort, UShort>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built
from the elements of `this` array and the [other] array with the same index\n * using the provided [transform]
function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UIntArray.zip(other: UIntArray, transform: (a: UInt, b: UInt) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ULongArray.zip(other: ULongArray, transform: (a: ULong, b: ULong) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByte) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
    }\n    return list\n}\n\n/**\n * Returns a list
of values built from the elements of `this` array and the [other] array with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n *
@sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UShortArray.zip(other: UShortArray, transform: (a: UShort, b: UShort) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
    }\n    return list\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Array<out UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this)
{\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out UShort>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n

```

```

sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all
elements in the array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sum(): UInt {\n    return storage.sum().toUInt()\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sum(): ULong {\n    return storage.sum().toULong()\n}\n\n/**\n * Returns the sum of all elements in
the array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UByteArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UCollectionsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an array of UByte containing all of the
elements of this collection.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Collection<UByte>.toUByteArray(): UByteArray {\n    val result = UByteArray(size)\n    var index = 0\n    for
(element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns an array of UInt containing
all of the elements of this collection.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Collection<UInt>.toUIntArray(): UIntArray {\n    val result = UIntArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns an
array of ULong containing all of the elements of this collection.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<ULong>.toULongArray():
ULongArray {\n    val result = ULongArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] =
element\n    return result\n}\n\n/**\n * Returns an array of UShort containing all of the elements of this collection.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UShort>.toUShortArray():
UShortArray {\n    val result = UShortArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] =
element\n    return result\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Iterable<UInt>.sum():
UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\nReturns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*/\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<UShort>.sum(): UInt {\n
var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n"/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UComparisonsKt")\n\npackage
kotlin.comparisons\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the
greater of two values.\n

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UInt, b:
UInt): UInt {\n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: ULong,
b: ULong): ULong {\n    return
if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UByte,
b: UByte): UByte {\n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UShort,
b: UShort): UShort {\n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns
the greater of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n *
Returns the greater of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UByte, b: UByte, c: UByte): UByte {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n *
Returns the greater of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UShort, b: UShort, c: UShort): UShort {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n
*\n * Returns the greater of the given values.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
maxOf(a: UInt, vararg other: UInt): UInt {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: ULong, vararg other: ULong):
ULong {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n * Returns the greater
of the
given values.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UByte, vararg
other: UByte): UByte {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n *
Returns the greater of the given values.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
maxOf(a: UShort, vararg other: UShort): UShort {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n
return max\n}\n\n/**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UInt, b:
UInt): UInt {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: ULong,
b: ULong): ULong {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun minOf(a: UByte, b: UByte): UByte {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two
values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
minOf(a: UShort, b: UShort): UShort {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of three
values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: UInt, b: UInt, c: UInt): UInt {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns
the smaller of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n *
Returns the smaller of three values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: UByte, b: UByte,

```



```

ExperimentalUnsignedTypes::class)
public fun ULongRange.randomOrNull(random: Random): ULong? {
    if (isEmpty())
        return null
    return random.nextULong(this)
}
// Returns `true` if this range contains the specified [element].
// Always returns `false` if the [element] is `null`.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline operator fun UIntRange.contains(element: UInt?): Boolean
{
    return element != null && contains(element)
}
// Returns `true` if this range contains the specified [element].
// Always returns `false` if the [element] is `null`.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline operator fun ULongRange.contains(element: ULong?): Boolean {
    return element != null && contains(element)
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun UIntRange.contains(value: UByte): Boolean {
    return contains(value.toInt())
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun ULongRange.contains(value: UByte): Boolean {
    return contains(value.toULong())
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun ULongRange.contains(value: UInt): Boolean {
    return contains(value.toULong())
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun UIntRange.contains(value: ULong): Boolean {
    return (value shr UInt.SIZE_BITS) == 0uL && contains(value.toInt())
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun UIntRange.contains(value: UShort): Boolean {
    return contains(value.toInt())
}
// Checks if the specified [value] belongs to this range.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public operator fun ULongRange.contains(value: UShort): Boolean {
    return contains(value.toULong())
}
// Returns a progression from this value down to the specified [to] value with the step -1.
// The [to] value should be less than or equal to `this` value.
// If the [to] value is greater than `this` value the returned progression is empty.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public infix fun UByte.downTo(to: UByte): UIntProgression {
    return UIntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
// Returns a progression from this value down to the specified [to] value with the step -1.
// The [to] value should be less than or equal to `this` value.
// If the [to] value is greater than `this` value the returned progression is empty.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public infix fun UInt.downTo(to: UInt): UIntProgression {
    return UIntProgression.fromClosedRange(this, to, -1)
}
// Returns a progression from this value down to the specified [to] value with the step -1.
// The [to] value should be less than or equal to `this` value.
// If the [to] value is greater than `this` value the returned progression is empty.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public infix fun ULong.downTo(to: ULong): ULongProgression {
    return ULongProgression.fromClosedRange(this, to, -1L)
}
// Returns a progression from this value down to the specified [to] value with the step -1.
// The [to] value should be less than or equal to `this` value.
// If the [to] value is greater than `this` value the returned progression is empty.
*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public infix fun UShort.downTo(to: UShort): UIntProgression {
    return UIntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
// Returns a progression that goes over the same range in the opposite direction with the

```

```

same step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun UIntProgression.reversed(): UIntProgression {\n    return UIntProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same
step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULongProgression.reversed(): ULongProgression {\n    return ULongProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UIntProgression.step(step: Int): UIntProgression {\n    checkStepIsPositive(step > 0, step)\n    return
UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression
that goes over the same range with the given step.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
infix fun ULongProgression.step(step: Long): ULongProgression {\n    checkStepIsPositive(step > 0, step)\n
return ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a range
from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this`
value, then the returned range is empty.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UByte.until(to: UByte): UIntRange {\n    if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY\n    return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UInt.until(to:
UInt): UIntRange {\n    if (to <= UInt.MIN_VALUE)
return UIntRange.EMPTY\n    return this .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to
but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
infix fun ULong.until(to: ULong): ULongRange {\n    if (to <= ULong.MIN_VALUE) return
ULongRange.EMPTY\n    return this .. (to - 1u).toULong()\n}\n\n/**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
infix fun UShort.until(to: UShort): UIntRange {\n    if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY\n
return this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Ensures that this value is not less than the specified
[minimumValue].\n
*/\n\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n *
\n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtLeast(minimumValue: UInt): UInt {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtLeast(minimumValue: ULong): ULong {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or
equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtLeast(minimumValue: UByte): UByte {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample

```

```

samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtLeast(minimumValue: UShort): UShort {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue]
otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtMost(maximumValue: UInt): UInt {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtMost(maximumValue: ULong): ULong {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtMost(maximumValue: UByte): UByte {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtMost(maximumValue: UShort): UShort {\n    return if (this > maximumValue) maximumValue
else this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n
* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or
[maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {\n    if (minimumValue > maximumValue)
throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this
value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n *
@sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {\n    if (minimumValue
> maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {\n    if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n
if (this

```

```

> maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified
range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this
value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n *
@sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {\n    if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the
specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is
less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n *
@sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(range: ClosedRange<UInt>): UInt {\n    if (range is ClosedFloatingPointRange) {\n        return
this.coerceIn<UInt>(range)\n    }\n    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to
an empty range: $range.")\n    return when {\n        this < range.start -> range.start\n        this > range.endInclusive -
> range.endInclusive\n        else -> this\n    }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or
`range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {\n    if (range is ClosedFloatingPointRange) {\n
return this.coerceIn<ULong>(range)\n    }\n    if (range.isEmpty()) throw IllegalArgumentException("Cannot
coerce value to an empty range: $range.")\n    return when {\n        this < range.start -> range.start\n        this >
range.endInclusive -> range.endInclusive\n        else -> this\n    }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("USequencesKt")\n\npackage
kotlin.sequences\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Sequence<UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this)
{\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n *
The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the
sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UShort>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin\n\npublic expect open class Error : Throwable {\n

```



```

*\n@SinceKotlin("1.4")\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic expect fun
Throwable.addSuppressed(exception: Throwable)\n\n/**\n * Returns a list of all exceptions that were suppressed in
order to deliver this exception.\n *\n * The list can be empty:\n * - if no exceptions were suppressed;\n * - if the
platform doesn't support suppressed exceptions;\n * - if this [Throwable] instance has
disabled the suppression.\n *\n@SinceKotlin("1.4")\npublic expect val Throwable.suppressedExceptions:
List<Throwable>\n","/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlin.js\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * Gives
a declaration (a function, a property or a class) specific name in JavaScript.\n *\n@Target(CLASS, FUNCTION,
PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)\n@OptionalExpectation\npublic
expect annotation class JsName(val name: String)\n\n/**\n * Marks experimental JS export annotations.\n *\n *
Note that behavior of these annotations will likely be changed in the future.\n *\n * Usages of such annotations will
be reported as warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g.
`@OptIn(ExperimentalJsExport::class)`,\n * or with the `-opt-in=kotlin.js.ExperimentalJsExport`
compiler option is given.\n *\n@RequiresOptIn(level =
RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Since
Kotlin("1.4")\npublic annotation class ExperimentalJsExport\n\n/**\n * Exports top-level declaration on JS
platform.\n *\n * Compiled module exposes declarations that are marked with this annotation without name
mangling.\n *\n * This annotation can be applied to either files or top-level declarations.\n *\n * It is currently
prohibited to export the following kinds of declarations:\n *\n * * `expect` declarations\n * * inline functions with
reified type parameters\n * * suspend functions\n * * secondary constructors without `@JsName`\n * *
extension properties\n * * enum classes\n * * annotation classes\n *\n * Signatures of exported declarations must
only contain "exportable" types:\n *\n * * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`,
`Double`\n * * `BooleanArray`,
`ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * * `Array<exportable-type>`\n * *
Function types with exportable parameters and return types\n * * `external` or `@JsExport` classes and interfaces\n
* * Nullable counterparts of types above\n * * Unit return type. Must not be nullable\n *\n * This annotation is
experimental, meaning that restrictions mentioned above are subject to change.\n
*\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY,
FUNCTION, FILE)\n@SinceKotlin("1.4")\n@OptionalExpectation\npublic expect annotation class
JsExport(),"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.io\n\n/** Prints the line separator to the standard output stream. *\npublic expect fun
println()\n\n/** Prints the given [message]
and the line separator to the standard output stream. *\npublic expect fun println(message: Any?)\n\n/** Prints the
given [message] to the standard output stream. *\npublic expect fun print(message: Any?)\n\n/** Reads a line
of input from the standard input stream and returns it,\n * or throws a [RuntimeException] if EOF has already been
reached when [readln] is called.\n *\n * LF or CRLF is treated as the line terminator. Line terminator is not included
in the returned string.\n *\n * Currently this function is not supported in Kotlin/JS and throws
[UnsupportedOperationException].\n *\n@SinceKotlin("1.6")\npublic expect fun readln(): String\n\n/** Reads a line of input from the standard input stream and returns it,\n * or return `null` if EOF has already been
reached when [readlnOrNull] is called.\n *\n * LF or CRLF is treated as the line terminator. Line terminator is not
included in the returned string.\n *\n * Currently this function is not supported in Kotlin/JS and throws
[UnsupportedOperationException].\n *\n@SinceKotlin("1.6")\npublic expect fun readlnOrNull():
String?\n\ninternal class ReadAfterEOFException(message: String?) : RuntimeException(message)\n\ninternal
expect interface Serializable\n","/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlin.collections\n\nimport kotlin.internal.PlatformDependent\n\n/**\n *

```

Classes that inherit from this interface can be represented as a sequence of elements that can be iterated over.

@param T the type of element being iterated over. The iterator is covariant in its element type.

```

public interface Iterable<out T> {
    /**
     * Returns an iterator over the elements of this object.
     */
    public operator fun iterator(): Iterator<T>
}

```

Classes that inherit from this interface can be represented as a sequence of elements that can be iterated over and that supports removing elements during iteration.

@param T the type of element being iterated over. The mutable iterator is invariant in its element type.

```

public interface MutableIterable<out T> : Iterable<T> {
    /**
     * Returns an iterator over the elements of this sequence that supports removing elements during iteration.
     */
    override fun iterator(): MutableIterator<T>
}

```

A generic collection of elements. Methods in this interface support only read-only access to the collection; read/write access is supported through the [MutableCollection] interface.

@param E the type of elements contained in the collection. The collection is covariant in its element type.

```

public interface Collection<out E> : Iterable<E> {
    // Query Operations
    /**
     * Returns the size of the collection.
     */
    public val size: Int

    /**
     * Returns `true` if the collection is empty (contains no elements), `false` otherwise.
     */
    public fun isEmpty(): Boolean

    /**
     * Checks if the specified element is contained in this collection.
     */
    public operator fun contains(element: @UnsafeVariance E): Boolean

    override fun iterator(): Iterator<E>

    // Bulk Operations
    /**
     * Checks if all elements in the specified collection are contained in this collection.
     */
    public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean

    /**
     * A generic collection of elements that supports adding and removing elements.
     */
    @param E the type of elements contained in the collection. The mutable collection is invariant in its element type.
    public interface MutableCollection<E> : Collection<E>,
        MutableIterable<E> {
        // Query Operations
        override fun iterator(): MutableIterator<E>

        // Modification Operations
        /**
         * Adds the specified element to the collection.
         *
         * @return `true` if the element has been added, `false` if the collection does not support duplicates and the element is already contained in the collection.
         */
        public fun add(element: E): Boolean

        /**
         * Removes a single instance of the specified element from this collection, if it is present.
         *
         * @return `true` if the element has been successfully removed; `false` if it was not present in the collection.
         */
        public fun remove(element: E): Boolean

        // Bulk Modification Operations
        /**
         * Adds all of the elements of the specified collection to this collection.
         *
         * @return `true` if any of the specified elements was added to the collection, `false` if the collection was not modified.
         */
        public fun addAll(elements: Collection<E>): Boolean

        /**
         * Removes all of this collection's elements that are also contained in the specified collection.
         *
         * @return `true` if any of the specified elements was removed from the collection, `false` if the collection was not modified.
         */
        public fun removeAll(elements: Collection<E>): Boolean

        /**
         * Retains only the elements in this collection that are contained in the specified collection.
         *
         * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
         */
        public fun retainAll(elements: Collection<E>): Boolean

        /**
         * Removes all elements from this collection.
         */
        public fun clear(): Unit
    }

    /**
     * A generic ordered collection of elements. Methods in this interface support only read-only access to the list; read/write access is supported through the [MutableList] interface.
     */
    @param E the type of elements contained in the list. The list is covariant in its element type.
    public interface List<out E> : Collection<E> {
        // Query Operations
        override val size: Int
        override fun isEmpty(): Boolean

        override fun contains(element: @UnsafeVariance E): Boolean
        override fun iterator(): Iterator<E>

        // Bulk Operations
        override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean

        // Positional Access Operations
        /**
         * Returns the element at the specified index in the list.
         */
        public operator fun get(index: Int): E

        // Search Operations
        /**
         * Returns the index of the first occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
         */
        public fun indexOf(element: @UnsafeVariance E): Int

        /**
         * Returns the index of the last occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
         */
        public fun lastIndexOf(element: @UnsafeVariance E): Int

        // List Iterators
        /**
         * Returns a list

```

iterator over the elements in this list (in proper sequence).\n

```
*/\n public fun listIterator(): ListIterator<E>\n\n /**\n  * Returns a list iterator over the elements in this list  
(in proper sequence), starting at the specified [index].\n  */\n public fun listIterator(index: Int):
```

```
ListIterator<E>\n\n // View\n /**\n  * Returns a view of the portion of this list between the specified  
[fromIndex] (inclusive) and [toIndex] (exclusive).\n  * The returned list is backed by this list, so non-structural  
changes in the returned list are reflected in this list, and vice-versa.\n  */\n  * Structural changes in the base list  
make the behavior of the view undefined.\n  */\n public fun subList(fromIndex: Int, toIndex: Int):
```

```
List<E>\n}\n\n/**\n * A generic ordered collection of elements that supports adding and removing elements.\n *  
@param E the type of elements contained in the list. The mutable list is invariant in its element type.\n */\npublic  
interface MutableList<E> : List<E>, MutableCollection<E> {\n
```

```
 // Modification Operations\n /**\n  * Adds the specified element to the end of this list.\n  */\n  * @return  
`true` because the list is always modified as the result of this operation.\n  */\n override fun add(element: E):  
Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n /**\n  * Adds  
all of the elements of the specified collection to the end of this list.\n  */\n  * The elements are appended in the  
order they appear in the [elements] collection.\n  */\n  * @return `true` if the list was changed as the result of the  
operation.\n  */\n override fun addAll(elements: Collection<E>): Boolean\n\n /**\n  * Inserts all of the  
elements of the specified collection [elements] into this list at the specified [index].\n  */\n  * @return `true` if the  
list was changed as the result of the operation.\n  */\n public fun addAll(index: Int, elements: Collection<E>):  
Boolean\n\n override fun removeAll(elements:
```

```
Collection<E>): Boolean\n\n override fun retainAll(elements: Collection<E>): Boolean\n\n override fun clear():  
Unit\n\n // Positional Access Operations\n /**\n  * Replaces the element at the specified position in this list  
with the specified element.\n  */\n  * @return the element previously at the specified position.\n  */\n public  
operator fun set(index: Int, element: E): E\n\n /**\n  * Inserts an element into the list at the specified [index].\n  */\n public fun add(index: Int, element: E): Unit\n\n /**\n  * Removes an element at the specified [index] from  
the list.\n  */\n  * @return the element that has been removed.\n  */\n public fun removeAt(index: Int): E\n\n // List Iterators\n override fun listIterator(): MutableListIterator<E>\n\n override fun listIterator(index: Int):  
MutableListIterator<E>\n\n // View\n override fun subList(fromIndex: Int, toIndex: Int):  
MutableList<E>\n}\n\n/**\n * A generic
```

```
unordered collection of elements that does not support duplicate elements.\n * Methods in this interface support  
only read-only access to the set;\n * read/write access is supported through the [MutableSet] interface.\n * @param  
E the type of elements contained in the set. The set is covariant in its element type.\n */\npublic interface Set<out E>  
: Collection<E> {\n // Query Operations\n\n override val size: Int\n\n override fun isEmpty(): Boolean\n\n override fun contains(element: @UnsafeVariance E): Boolean\n\n override fun iterator(): Iterator<E>\n\n // Bulk  
Operations\n\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n}\n\n/**\n * A  
generic unordered collection of elements that does not support duplicate elements, and supports\n * adding and  
removing elements.\n * @param E the type of elements contained in the set. The mutable set is invariant in its  
element type.\n */\npublic interface MutableSet<E> : Set<E>, MutableCollection<E> {\n
```

```
 // Query Operations\n\n override fun iterator(): MutableIterator<E>\n\n // Modification Operations\n\n /**\n  * Adds the specified element to the set.\n  */\n  * @return `true` if the element has been added, `false` if the  
element is already contained in the set.\n  */\n override fun add(element: E): Boolean\n\n override fun  
remove(element: E): Boolean\n\n // Bulk Modification Operations\n\n override fun addAll(elements:  
Collection<E>): Boolean\n\n override fun removeAll(elements: Collection<E>): Boolean\n\n override fun  
retainAll(elements: Collection<E>): Boolean\n\n override fun clear(): Unit\n}\n\n/**\n * A collection that holds  
pairs of objects (keys and values) and supports efficiently retrieving\n * the value corresponding to each key. Map  
keys are unique; the map holds only one value for each key.\n * Methods in this interface support only read-only  
access to the map; read-write access is supported through\n * the [MutableMap] interface.\n * @param  
K the type of map keys. The map is invariant in its key type, as it\n * can accept key as a parameter (of  
[containsKey] for example) and return it in [keys] set.\n * @param V the type of map values. The map is covariant
```



```

Iterator<Short> {\n  override final fun next() = nextShort()\n\n  /** Returns the next value in the sequence without
boxing. */\n  public abstract fun nextShort(): Short\n}\n\n/** An iterator over a sequence of values of type `Int`.\n*/\npublic abstract class IntIterator : Iterator<Int> {\n  override final fun next() = nextInt()\n\n  /** Returns the
next value in the sequence without boxing. */\n  public
  abstract fun nextInt(): Int\n}\n\n/** An iterator over a sequence of values of type `Long`.\n*/\npublic abstract class
LongIterator : Iterator<Long> {\n  override final fun next() = nextLong()\n\n  /** Returns the next value in the
sequence without boxing. */\n  public abstract fun nextLong(): Long\n}\n\n/** An iterator over a sequence of
values of type `Float`.\n*/\npublic abstract class FloatIterator : Iterator<Float> {\n  override final fun next() =
nextFloat()\n\n  /** Returns the next value in the sequence without boxing. */\n  public abstract fun nextFloat():
Float\n}\n\n/** An iterator over a sequence of values of type `Double`.\n*/\npublic abstract class DoubleIterator :
Iterator<Double> {\n  override final fun next() = nextDouble()\n\n  /** Returns the next value in the sequence
without boxing. */\n  public abstract fun nextDouble(): Double\n}\n\n/** An iterator over a sequence of values of
type `Boolean`.\n*/\npublic abstract class BooleanIterator : Iterator<Boolean>
{\n  override final fun next() = nextBoolean()\n\n  /** Returns the next value in the sequence without boxing. */\n
  public abstract fun nextBoolean(): Boolean\n}\n\n"/\n\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin.ranges\n\n/**\n * An iterator over a progression of values of type `Char`.\n\n * @property step the number by
which the value is incremented on each step.\n\n * @internal class CharProgressionIterator(first: Char, last: Char, val
step: Int) : CharIterator() {\n  private val finalElement: Int = last.code\n  private var hasNext: Boolean = if (step >
0) first <= last else first >= last\n  private var next: Int = if (hasNext) first.code else finalElement\n\n  override fun
hasNext(): Boolean = hasNext\n\n  override fun nextChar(): Char {\n
    val value = next\n    if (value == finalElement) {\n      if (!hasNext) throw
      kotlin.NoSuchElementException()\n      hasNext = false\n    }\n    else {\n      next += step\n    }\n
    return value.toChar()\n  }\n}\n\n/**\n * An iterator over a progression of values of type `Int`.\n\n * @property step
the number by which the value is incremented on each step.\n\n * @internal class IntProgressionIterator(first: Int, last:
Int, val step: Int) : IntIterator() {\n  private val finalElement: Int = last\n  private var hasNext: Boolean = if (step >
0) first <= last else first >= last\n  private var next: Int = if (hasNext) first else finalElement\n\n  override fun
hasNext(): Boolean = hasNext\n\n  override fun nextInt(): Int {\n    val value = next\n    if (value ==
    finalElement) {\n      if (!hasNext) throw kotlin.NoSuchElementException()\n      hasNext = false\n    }\n
    else {\n      next += step\n    }\n    return value\n  }\n}\n\n/**\n * An iterator over a progression of values of
type `Long`.\n\n * @property step the number by which the value is incremented on each step.\n\n * @internal class
LongProgressionIterator(first: Long, last: Long, val step: Long) : LongIterator() {\n  private val finalElement: Long =
last\n  private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n  private var next: Long = if
(hasNext) first else finalElement\n\n  override fun hasNext(): Boolean = hasNext\n\n  override fun nextLong(): Long {\n
    val value = next\n    if (value == finalElement) {\n      if (!hasNext) throw kotlin.NoSuchElementException()\n
      hasNext = false\n    }\n    else {\n      next += step\n    }\n    return value\n  }\n}\n\n"/\n\n *
Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n\n * Use of this source code is
governed by the Apache 2.0 license that can be found
in the license/LICENSE.txt file.\n\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport
kotlin.internal.getProgressionLastElement\n\n/**\n * A progression of values of type `Char`.\n\n * @public open class
CharProgression\n  internal constructor\n  (\n    start: Char,\n    endInclusive: Char,\n    step: Int\n  ) : Iterable<Char> {\n  init {\n    if (step == 0) throw kotlin.IllegalArgumentException("\u0027Step must be non-zero.\u0027")\n    if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("\u0027Step must be greater than Int.MIN_VALUE to avoid overflow on negation.\u0027")\n  }\n\n  /**\n   * The first element in the progression.\n\n   */\n  public val first: Char = start\n\n  /**\n   * The last element in the progression.\n\n   */\n  public val last: Char = getProgressionLastElement(start.code, endInclusive.code, step).toChar()\n\n  /**\n   * The step of the

```

```

progression.\n    */\n    public val step: Int
    = step\n\n    override fun iterator(): CharIterator = CharProgressionIterator(first, last, step)\n\n    /**\n     * Checks if
the progression is empty.\n     *\n     * Progression with a positive step is empty if its first element is greater than the
last element.\n     * Progression with a negative step is empty if its first element is less than the last element.\n     */\n
    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n    override fun equals(other:
Any?): Boolean =\n        other is CharProgression && (isEmpty() && other.isEmpty()) ||\n        first == other.first
&& last == other.last && step == other.step)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 *
(31 * first.code + last.code) + step)\n\n    override fun toString(): String = if (step > 0) \"$first..$last step $step\" else
\"$first downTo $last step ${-step}\"\n\n    companion object {\n        /**\n         * Creates CharProgression within
the specified bounds
of a closed range.\n         *\n         * The progression starts with the [rangeStart] value and goes toward the
[rangeEnd] value not excluding it, with the specified [step].\n         * In order to go backwards the [step] must be
negative.\n         *\n         * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n         */\n
        public fun fromClosedRange(rangeStart: Char, rangeEnd: Char, step: Int): CharProgression =
CharProgression(rangeStart, rangeEnd, step)\n    }\n\n    /**\n     * A progression of values of type `Int`.\n     */\n\n    public
open class IntProgression\n        internal constructor(\n            start: Int,\n            endInclusive: Int,\n            step:
Int\n        ): Iterable<Int> {\n        init {\n            if (step == 0) throw kotlin.IllegalArgumentException(\"Step must be non-
zero.\")\n            if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be greater than
Int.MIN_VALUE to avoid overflow on negation.\")\n        }\n\n        /**\n         * The first element in the progression.\n         */\n        public val first: Int = start\n\n        /**\n         * The last element in
the progression.\n         */\n        public val last: Int = getProgressionLastElement(start, endInclusive, step)\n\n        /**\n         *
The step of the progression.\n         */\n        public val step: Int = step\n\n        override fun iterator(): IntIterator =
IntProgressionIterator(first, last, step)\n\n        /**\n         * Checks if the progression is empty.\n         *\n         * Progression
with a positive step is empty if its first element is greater than the last element.\n         * Progression with a negative
step is empty if its first element is less than the last element.\n         */\n        public open fun isEmpty(): Boolean = if
(step > 0) first > last else first < last\n\n        override fun equals(other: Any?): Boolean =\n            other is IntProgression
&& (isEmpty() && other.isEmpty()) ||\n            first == other.first && last == other.last && step == other.step)\n\n        override fun hashCode():
Int =\n            if (isEmpty()) -1 else (31 * (31 * first + last) + step)\n\n        override fun toString(): String = if (step > 0)
\"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n        companion object {\n            /**\n             *
Creates IntProgression within the specified bounds of a closed range.\n             *\n             * The progression starts with
the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n             * In
order to go backwards the [step] must be negative.\n             *\n             * [step] must be greater than `Int.MIN_VALUE`
and not equal to zero.\n             */\n
            public fun fromClosedRange(rangeStart: Int, rangeEnd: Int, step: Int):
IntProgression = IntProgression(rangeStart, rangeEnd, step)\n        }\n\n        /**\n         * A progression of values of type
`Long`.\n         */\n\n        public open class LongProgression\n            internal constructor(\n                start: Long,\n                endInclusive: Long,\n                step: Long\n            ): Iterable<Long> {\n            init {\n                if (step == 0L) throw kotlin.IllegalArgumentException(\"Step
must be non-zero.\")\n                if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be
greater than Long.MIN_VALUE to avoid overflow on negation.\")\n            }\n\n            /**\n             * The first element in the
progression.\n             */\n            public val first: Long = start\n\n            /**\n             * The last element in the progression.\n             */\n            public val last: Long = getProgressionLastElement(start, endInclusive, step)\n\n            /**\n             * The step of the
progression.\n             */\n            public val step: Long = step\n\n            override fun iterator(): LongIterator =
LongProgressionIterator(first, last, step)\n\n            /**\n             * Checks if the progression is empty.\n             *\n             *
Progression with a positive step is empty if its first element is greater than the last element.\n             * Progression with a
negative step is empty if its first element is less than the last element.\n             */\n            public open fun isEmpty(): Boolean = if
(step > 0) first > last else first < last\n\n            override fun equals(other: Any?): Boolean =\n                other is LongProgression && (isEmpty() && other.isEmpty()) ||\n                first ==

```

```

other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n    if (isEmpty()) -1\n    else (31 * (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))) + (step xor (step ushr 32))).toInt()\n\n override\nfun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\""\n\n\ncompanion object {\n    /**\n     * Creates LongProgression within the specified bounds of a closed range.\n     *\n     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it,\n    with the specified [step].\n     * In order to go backwards the [step] must be negative.\n     * [step] must\n    be greater than `Long.MIN_VALUE`\n    and not equal to zero.\n     */\n    public fun fromClosedRange(rangeStart: Long, rangeEnd: Long, step: Long):\n    LongProgression = LongProgression(rangeStart, rangeEnd, step)\n}\n\n\n"/**\n * Copyright 2010-2019\n    JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the\n    Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.ranges\n\n/**\n * Represents a range of values (for example, numbers or characters).\n * See the [Kotlin language\n    documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\n\npublic interface\n    ClosedRange<T: Comparable<T>> {\n    /**\n     * The minimum value in the range.\n     */\n    public val start:\n    T\n\n    /**\n     * The maximum value in the range (inclusive).\n     */\n    public val endInclusive: T\n\n    /**\n     * Checks whether the specified [value] belongs to the range.\n     */\n    public operator fun contains(value:\n    T): Boolean = value >= start && value <= endInclusive\n\n    /**\n     * Checks whether the range is empty.\n     *\n     * The range is empty if its start value is greater than the end value.\n     */\n    public fun isEmpty(): Boolean =\n    start > endInclusive\n}\n\n\n"/**\n * Copyright 2010-2015 JetBrains s.r.o.\n * Licensed under the Apache License,\n    Version 2.0 (the \"License\");\n * you may not use this file except in compliance with the License.\n * You may\n    obtain a copy of the License at\n * http://www.apache.org/licenses/LICENSE-2.0\n * Unless required by applicable law or agreed to in writing,\n    software\n * distributed under the License is distributed on an \"AS IS\"\n    BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the\n    License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage\n    kotlin\n\n/**\n * The type with only one value: the `Unit` object. This type corresponds to the\n    `void` type in Java.\n */\n\npublic object Unit {\n    override fun toString() = \"kotlin.Unit\"\n}\n\n\n"/**\n * Copyright\n    2010-2015 JetBrains s.r.o.\n * Licensed under the Apache License, Version 2.0 (the \"License\");\n * you may\n    not use this file except in compliance with the License.\n * You may obtain a copy of the License at\n * http://www.apache.org/licenses/LICENSE-2.0\n * Unless required by applicable law or agreed to in writing,\n    software\n * distributed under the License is distributed on an \"AS IS\" BASIS,\n * WITHOUT WARRANTIES\n    OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language\n    governing permissions and\n * limitations under the License.\n */\n\npackage kotlin.annotation\n\nimport\n    kotlin.annotation.AnnotationTarget\n\n/**\n * Contains the list of code elements which are the possible annotation\n    targets\n */\n\npublic enum class AnnotationTarget {\n    /**\n     * Class, interface or object, annotation class is also\n    included\n     *\n     * CLASS,\n     */\n    /**\n     * Annotation class only\n     */\n    ANNOTATION_CLASS,\n    /**\n     * Generic type parameter\n     */\n    TYPE_PARAMETER,\n    /**\n     * Property\n     */\n    PROPERTY,\n    /**\n     * Field, including property's backing field\n     */\n    FIELD,\n    /**\n     * Local variable\n     */\n    LOCAL_VARIABLE,\n    /**\n     * Value parameter of a function or a constructor\n     */\n    VALUE_PARAMETER,\n    /**\n     * Constructor only (primary or secondary)\n     */\n    CONSTRUCTOR,\n    /**\n     * Function (constructors are not included)\n     */\n    FUNCTION,\n    /**\n     * Property getter only\n     */\n    PROPERTY_GETTER,\n    /**\n     * Property setter only\n     */\n    PROPERTY_SETTER,\n    /**\n     * Type usage\n     */\n    TYPE,\n    /**\n     * Any expression\n     */\n    EXPRESSION,\n    /**\n     * File\n     */\n    FILE,\n    /**\n     * Type alias\n     */\n    TYPEALIAS\n}\n\n\n/**\n * Contains the list of possible annotation's retentions.\n *\n * Determines how an annotation is stored in binary output.\n */\n\npublic enum class AnnotationRetention {\n    /**\n     * Annotation isn't stored in binary\n    output\n     */\n    SOURCE,\n    /**\n     * Annotation is stored in binary output, but invisible for reflection\n     */\n    BINARY,\n    /**\n     * Annotation is stored in binary output and visible for reflection (default retention)\n     */\n    RUNTIME\n}\n\n\n/**\n * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation.\n *\n * If

```

the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:\n * [CLASS], [PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER], [PROPERTY_SETTER].\n *\n * @property allowedTargets list of allowed annotation targets\n

```

*\n*@Target(AnnotationTarget.ANNOTATION_CLASS)\n@MustBeDocumented\npublic annotation class Target(vararg val allowedTargets: AnnotationTarget)\n\n/**\n * This meta-annotation determines whether an annotation is stored in binary output and visible for reflection. By default, both are true.\n *\n * @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)\n
```

```

*\n*@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Retention(val value: AnnotationRetention = AnnotationRetention.RUNTIME)\n\n/**\n * This meta-annotation determines that an annotation is applicable twice or more on a single code element\n
```

```

*\n*@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Repeatable\n\n/**\n * This meta-annotation determines that an annotation is a part of public API and therefore should be included in the generated\n * documentation for the element to which the annotation is applied.\n
```

```

*\n*@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class MustBeDocumented\n"/*\n * Copyright 2010-2016 JetBrains s.r.o.\n *\n * Licensed under the Apache License, Version 2.0 (the "License");\n *\n * you may not use this file except in compliance with the License.\n *\n * You may obtain a copy of the License at\n *\n * http://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by applicable law or agreed to in writing, software\n * distributed under the License is distributed on an\n * "AS IS" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks\n * That means it's completely safe to use generic types as argument for such parameter.\n
```

```

*\n*@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PureReifiable\n\n/**\n * Specifies that the corresponding built-in method exists depending on platform.\n * Current implementation for JVM looks whether method with same JVM descriptor exists in the module JDK.\n * For example MutableMap.remove(K, V) available only if corresponding\n * method 'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)' is defined in JDK (i.e. for major versions >= 8)\n
```

```

*\n*@Target(AnnotationTarget.FUNCTION)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PlatformDependent\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n// a mod b (in arithmetical sense)\nprivate fun mod(a: Int, b: Int): Int {\n    val mod = a % b\n    return if (mod >= 0) mod else mod + b\n}\n\nprivate fun mod(a: Long, b: Long): Long {\n    val mod = a % b\n    return if (mod >= 0) mod else mod + b\n}\n\n// (a - b) mod c\nprivate fun differenceModulo(a: Int, b: Int, c: Int): Int {\n    return mod(mod(a, c) - mod(b, c), c)\n}\n\nprivate fun differenceModulo(a: Long, b: Long, c: Long): Long {\n    return mod(mod(a, c) - mod(b, c), c)\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress\n */\n\n@PublishedApi\ninternal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)\n    else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n\n/**\n
```

```

* Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is
in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n *
[step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the
condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start
first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or
difference of successive elements in the progression\n * @return the final element of the progression\n *
@suppress\n */\n@PublishedApi\ninternal fun getProgressionLastElement(start: Long, end: Long, step: Long):
Long = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n    step < 0 -> if
(start <= end) end else end + differenceModulo(start, end, -step)\n    else -> throw
kotlin.IllegalArgumentException("Step is zero.")\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n@JsName("arrayIterator")\ninternal fun arrayIterator(array:
dynamic, type: String?) = when (type) {\n    null -> {\n        val arr: Array<dynamic> = array\n        object :
Iterator<dynamic> {\n            var index = 0\n            override fun hasNext() = index < arr.size\n            override fun
next() = if (index < arr.size) arr[index++] else throw NoSuchElementException("$index")\n        }\n    }\n    "BooleanArray" -> booleanArrayIterator(array)\n    "ByteArray" -> byteArrayIterator(array)\n    "ShortArray" -
> shortArrayIterator(array)\n    "CharArray" -> charArrayIterator(array)\n    "IntArray" ->
intArrayIterator(array)\n    "LongArray" -> longArrayIterator(array)\n    "FloatArray" ->
floatArrayIterator(array)\n    "DoubleArray" -> doubleArrayIterator(array)\n    else -> throw IllegalStateException("Unsupported type
argument for arrayIterator: $type")\n}\n\n@JsName("booleanArrayIterator")\ninternal fun
booleanArrayIterator(array: BooleanArray) = object : BooleanIterator() {\n    var index = 0\n    override fun
hasNext() = index < array.size\n    override fun nextBoolean() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("byteArrayIterator")\ninternal fun byteArrayIterator(array:
ByteArray) = object : ByteIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override
fun nextByte() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("shortArrayIterator")\ninternal fun
shortArrayIterator(array: ShortArray) = object : ShortIterator() {\n    var index = 0\n    override fun hasNext() =
index < array.size\n    override fun nextShort()
= if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("charArrayIterator")\ninternal fun charArrayIterator(array:
CharArray) = object : CharIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override
fun nextChar() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("intArrayIterator")\ninternal fun intArrayIterator(array:
IntArray) = object : IntIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override fun
nextInt() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("floatArrayIterator")\ninternal fun
floatArrayIterator(array: FloatArray) = object : FloatIterator() {\n    var index = 0\n    override fun hasNext() = index
< array.size\n    override fun nextFloat() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("doubleArrayIterator")\ninternal
fun doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {\n    var index = 0\n    override fun
hasNext() = index < array.size\n    override fun nextDouble() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("longArrayIterator")\ninternal fun longArrayIterator(array:
LongArray) = object : LongIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override
fun nextLong() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("PropertyMetadata")\ninternal class
PropertyMetadata(@JsName("callableName") val name:
String)\n\n@JsName("noWhenBranchMatched")\ninternal fun noWhenBranchMatched(): Nothing = throw

```

```

NoWhenBranchMatchedException()\n\n@JsName("subSequence")\ninternal fun subSequence(c: CharSequence,
startIndex: Int, endIndex: Int): CharSequence {\n    if (c is String) {\n        return c.substring(startIndex,
endIndex)\n    } else {\n        return c.asDynamic().`subSequence_vux9f0$`(startIndex, endIndex)\n
}\n}\n\n@JsName("captureStack")\ninternal fun captureStack(@Suppress("UNUSED_PARAMETER")
baseClass: JsClass<in Throwable>, instance: Throwable) {\n    if (js("Error").captureStackTrace) {\n        // Using
uncropped stack traces due to KT-37563.\n        // Precise stack traces are implemented in JS IR compiler and
stdlib\n        js("Error").captureStackTrace(instance);\n    } else {\n        instance.asDynamic().stack = js("new
Error()").stack;\n    }\n}\n\n@JsName("newThrowable")\ninternal fun newThrowable(message: String?, cause:
Throwable?): Throwable {\n    val throwable = js("new Error()")\n    throwable.message = if (jsTypeOf(message)
== "undefined") {\n        if (cause != null) cause.toString() else null\n    } else {\n        message\n    }\n
throwable.cause = cause\n    throwable.name = "Throwable"\n    return
throwable\n}\n\n@JsName("BoxedChar")\ninternal
class BoxedChar(val c: Int) : Comparable<Int> {\n    override fun equals(other: Any?): Boolean {\n        return
other is BoxedChar && c == other.c\n    }\n\n    override fun hashCode(): Int {\n        return c\n    }\n\n    override
fun toString(): String {\n        return js("this.c").unsafeCast<Char>().toString()\n    }\n\n    override fun
compareTo(other: Int): Int {\n        return js("this.c - other").unsafeCast<Int>()\n    }\n\n
@JsName("valueOf")\n    public fun valueOf(): Int {\n        return c\n    }\n}\n\n@kotlin.internal.InlineOnly\ninternal inline fun <T> concat(args: Array<T>): T {\n    val typed =
js("Array")(args.size)\n    for (i in args.indices) {\n        val arr = args[i]\n        if (arr !is Array<*>) {\n
typed[i] = js("[]").slice.call(arr)\n        } else {\n            typed[i] = arr\n        }\n    }\n    return
js("[]").concat.apply(js("[]"), typed);\n}\n\n/** Concat regular Array's and TypedArray's into an Array.\n
*/\n\n@PublishedApi\n@JsName("arrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal
fun <T> arrayConcat(a: T, b: T): T {\n    return concat(js("arguments"))\n}\n\n/** Concat primitive arrays. Main
use: prepare vararg arguments.\n * For compatibility with 1.1.0 the arguments may be a mixture of Array's and
TypedArray's.\n * If the first argument is TypedArray (Byte-, Short-, Char-, Int-, Float-, and DoubleArray)
returns a TypedArray, otherwise an Array.\n * If the first argument has the $type$ property (Boolean-, Char-, and
LongArray) copy its value to result.$type$.\n * If the first argument is a regular Array without the $type$ property
default to arrayConcat.\n
*/\n\n@PublishedApi\n@JsName("primitiveArrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal
fun <T> primitiveArrayConcat(a: T, b: T): T {\n    val args: Array<T> = js("arguments")\n    if (a is Array<*> &&
a.asDynamic().`$type$` === undefined) {\n        return concat(args)\n    } else {\n        var size = 0\n
        for (i in args.indices) {\n            size += args[i].asDynamic().length as Int\n        }\n        val result = js("new
a.constructor(size)")\n        kotlin.copyArrayType(a, result)\n        size = 0\n        for (i in args.indices) {\n            val
arr = args[i].asDynamic()\n            for (j in 0 until arr.length) {\n                result[size++] = arr[j]\n            }\n
        }\n        return result\n    }\n}\n\n@JsName("booleanArrayOf")\ninternal fun booleanArrayOf() =
withType("BooleanArray", js("[]").slice.call(arguments))\n}\n\n@JsName("charArrayOf") // The arguments have
to be slice'd here because of Rhino (see KT-16974)\ninternal fun charArrayOf() = withType("CharArray", js("new
Uint16Array([]).slice.call(arguments)))\n}\n\n@JsName("longArrayOf")\ninternal fun longArrayOf() =
withType("LongArray",
js("[]").slice.call(arguments))\n}\n\n@JsName("withType")\n@kotlin.internal.InlineOnly\ninternal inline fun
withType(type: String, array: dynamic):
dynamic {\n    array.`$type$` = type\n    return array\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Function corresponding to JavaScript's
`typeof` operator\n */\n\n@kotlin.internal.InlineOnly\n@Suppress("UNUSED_PARAMETER")\npublic inline fun
jsTypeOf(a: Any?): String = js("typeof a")\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n@file:Suppress("UNUSED_PARAMETER",

```

```

\["NOTHING_TO_INLINE"]\n\npackage kotlin\n\n/**\n * Returns an empty array of the specified type [T].\n *\npublic inline fun <T> emptyArray(): Array<T> = js{""}\n\n@library\npublic fun <T> arrayOf(vararg elements: T): Array<T> = definedExternally\n\n@library\npublic fun doubleArrayOf(vararg elements: Double): DoubleArray = definedExternally\n\n@library\npublic fun floatArrayOf(vararg elements: Float): FloatArray = definedExternally\n\n@library\npublic fun longArrayOf(vararg elements: Long): LongArray = definedExternally\n\n@library\npublic fun intArrayOf(vararg elements: Int): IntArray = definedExternally\n\n@library\npublic fun charArrayOf(vararg elements: Char): CharArray = definedExternally\n\n@library\npublic fun shortArrayOf(vararg elements: Short): ShortArray = definedExternally\n\n@library\npublic fun byteArrayOf(vararg elements: Byte): ByteArray = definedExternally\n\n@library\npublic fun booleanArrayOf(vararg elements: Boolean): BooleanArray = definedExternally\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\npublic actual fun <T> lazy(initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\n * The [mode] parameter is ignored.\n *\npublic actual fun <T> lazy(mode: LazyThreadSafetyMode, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\n * The [lock] parameter is ignored.\n *\npublic actual fun <T> lazy(lock: Any?, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n\ninternal fun fillFrom(src: dynamic, dst: dynamic): dynamic {\n    val srcLen: Int = src.length\n    val dstLen: Int = dst.length\n    var index: Int = 0\n    while (index < srcLen && index < dstLen)\n        dst[index] = src[index++]\n    return dst\n}\n\n\ninternal fun arrayCopyResize(source: dynamic, newSize: Int, default\Value: Any?): dynamic {\n    val result = source.slice(0, newSize)\n    copyArrayType(source, result)\n    var index: Int = source.length\n    if (newSize > index) {\n        result.length = newSize\n        while (index < newSize)\n            result[index++] = default\Value\n    }\n    return result\n}\n\n\ninternal fun <T> arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {\n    val result = array.slice()\n    result.length += collection.size\n    copyArrayType(array, result)\n    var index: Int = array.length\n    for (element in collection)\n        result[index++] = element\n    return result\n}\n\n\ninternal fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>): dynamic {\n    var index = startIndex\n    for (element in collection)\n        dst[index++] = element\n    return dst\n}\n\n\ninternal inline fun copyArrayType(from: dynamic, to: dynamic) {\n    if (from.`$type$` !== undefined) {\n        to.`$type$` = from.`$type$`\n    }\n}\n\n\ninternal inline fun jsIsType(obj: dynamic, jsClass: dynamic) = js{""Kotlin""}.isType(obj, jsClass), /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin\n\n/**\n * Creates a Char with the specified [code].\n *\n * @sample samples.text.Chars.charFromCode\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Char(code: UShort): Char {\n    return code.toInt().toChar()\n}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.coroutines\n\nimport kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@SinceKotlin("1.3")\n@JsName("CoroutineImpl")\n\ninternal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?> {\n    protected var state = 0\n    protected var exceptionState = 0\n    protected var result: Any? = null\n    protected var exception: Throwable? = null\n    protected var finallyPath: Array<Int?> = null\n\n    public override val context: CoroutineContext = resultContinuation.context\n\n    private var intercepted_: Continuation<Any?>? = null\n\n    public fun intercepted(): Continuation<Any?> =\n        intercepted_ ?: (context[ContinuationInterceptor]?.interceptContinuation(this) ?: this)\n        .also { intercepted_ = it }\n\n    override fun resumeWith(result: Result<Any?>) {\n        var current = this\n        var currentResult: Any? = result.getOrNull()\n        var currentException: Throwable? = result.exceptionOrNull()\n        // This loop unrolls recursion in current.resumeWith(param) to make saner and shorter stack traces on resume\n        while (true) {\n

```



```

with(current) { \n          val completion = resultContinuation \n \n          // Set result and exception fields in
the current continuation \n          if (currentException == null) { \n
    this.result = currentResult \n          } else { \n          state = exceptionState \n          exception =
currentException \n          } \n \n          try { \n          val outcome = doResume() \n          if (outcome
=== COROUTINE_SUSPENDED) return \n          currentResult = outcome \n          currentException =
null \n          } catch (exception: dynamic) { // Catch all exceptions \n          currentResult = null \n
currentException = exception.unsafeCast<Throwable>() \n          } \n \n          releaseIntercepted() // this state
machine instance is terminating \n \n          if (completion is CoroutineImpl) { \n          // unrolling recursion
via loop \n          current = completion \n          } else { \n          // top-level completion reached --
invoke and return \n          currentException?.let { \n
    completion.resumeWithException(it) \n          } ?: completion.resume(currentResult) \n
return \n          } \n          } \n          } \n          } \n \n          private fun releaseIntercepted() { \n          val intercepted =
intercepted_ \n          if (intercepted != null && intercepted !== this) { \n
context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted) \n          } \n          this.intercepted_ =
CompletedContinuation // just in case \n          } \n \n          protected abstract fun doResume(): Any? \n          } \n \n          internal object
CompletedContinuation : Continuation<Any?> { \n          override val context: CoroutineContext \n          get() =
error("This continuation is already complete") \n \n          override fun resumeWith(result: Result<Any?>) { \n
error("This continuation is already complete") \n          } \n \n          override fun toString(): String = "This continuation is
already complete" \n          } \n \n          /* \n          * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n          * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file. \n          * \n          * @file: Suppress("UNCHECKED_CAST",
"RedundantVisibilityModifier") \n          * \n          * package kotlin \n          * \n          * import kotlin.contracts.* \n          * \n          * import
kotlin.internal.InlineOnly \n          * \n          * import kotlin.jvm.JvmField \n          * \n          * import kotlin.jvm.JvmInline \n          * \n          * import
kotlin.jvm.JvmName \n          * \n          * /** \n          * A discriminated union that encapsulates a successful outcome with a value of type
[T] \n          * or a failure with an arbitrary [Throwable] exception. \n          * \n          * @SinceKotlin("1.3") \n          * @JvmInline \n          * public
value class Result<out T> @PublishedApi internal constructor( \n          * @PublishedApi \n          * internal val value: Any? \n          * ) :
Serializable { \n          * // discovery \n          * /** \n          * Returns `true` if this instance represents a successful outcome. \n          *
In this case [isFailure] returns `false`. \n          * \n          * @ \n          * public val isSuccess: Boolean get() = value !is Failure \n          *
/** \n          * Returns `true` if this instance represents
a failed outcome. \n          * In this case [isSuccess] returns `false`. \n          * \n          * @ \n          * public val isFailure: Boolean get() =
value is Failure \n          * // value & exception retrieval \n          * /** \n          * Returns the encapsulated value if this instance
represents [success][Result.isSuccess] or `null` \n          * if it is [failure][Result.isFailure]. \n          * \n          * This function is
a shorthand for `getOrElse { null }` (see [getOrElse]) or \n          * `fold(onSuccess = { it }, onFailure = { null })` (see
[fold]). \n          * \n          * @InlineOnly \n          * public inline fun getOrNull(): T? = \n          * when { \n          * isFailure -> null \n
else -> value as T \n          * } \n          * /** \n          * Returns the encapsulated [Throwable] exception if this instance
represents [failure][isFailure] or `null` \n          * if it is [success][isSuccess]. \n          * \n          * This function is a shorthand
for `fold(onSuccess = { null }, onFailure = { it })` (see [fold]). \n          * \n          * @ \n          * public fun exceptionOrNull(): Throwable?
= \n          * when (value) { \n          * is Failure -> value.exception \n          * else -> null \n          * } \n          * /** \n          * Returns a string
`Success(v)` if this instance represents [success][Result.isSuccess] \n          * where `v` is a string representation of the
value or a string `Failure(x)` \n          * if it is [failure][isFailure] where `x` is a string representation of the exception. \n          *
\ n          * @ \n          * public override fun toString(): String = \n          * when (value) { \n          * is Failure -> value.toString() //
"Failure($exception)" \n          * else -> "Success($value)" \n          * } \n          * // companion with constructors \n          * /** \n          *
Companion object for [Result] class that contains its constructor functions \n          * [success] and [failure]. \n          * \n          * @ \n          *
public companion object { \n          * /** \n          * Returns an instance that encapsulates the given [value] as successful
value. \n          * \n          * @ \n          * @Suppress("INAPPLICABLE_JVM_NAME") \n          * @InlineOnly \n          * @JvmName("success") \n

```

```

public inline fun <T> success(value: T): Result<T> =\n        Result(value)\n\n        /**\n        * Returns an
instance that encapsulates the given [Throwable] [exception] as failure.\n        */\n
@Suppress("INAPPLICABLE_JVM_NAME")\n        @InlineOnly\n        @JvmName("failure")\n        public
inline fun <T> failure(exception: Throwable): Result<T> =\n        Result(createFailure(exception))\n    }\n\n
internal class Failure(\n        @JvmField\n        val exception: Throwable\n    ): Serializable {\n        override fun
equals(other: Any?): Boolean = other is Failure && exception == other.exception\n        override fun hashCode():
Int = exception.hashCode()\n        override fun toString(): String = "Failure($exception)"\n    }\n\n/**\n *
Creates an instance of internal marker [Result.Failure] class to\n * make sure that this class is not exposed in ABI.\n
*/\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun createFailure(exception: Throwable):
Any =\n    Result.Failure(exception)\n\n/**\n * Throws exception if the result is failure. This internal function
minimizes\n * inlined bytecode for [getOrThrow] and makes sure that in the future we can\n * add some exception-
augmenting logic here (if needed).\n */\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun
Result<*>.throwOnFailure() {\n    if (value is Result.Failure) throw value.exception\n}\n\n/**\n * Calls the
specified function [block] and returns its encapsulated result if invocation was successful,\n * catching any
[Throwable] exception that was thrown from the [block] function execution and encapsulating it as a failure.\n
*/\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R> runCatching(block: () -> R): Result<R> {\n
return try {\n    Result.success(block())\n    } catch (e: Throwable) {\n    Result.failure(e)\n    }\n}\n\n/**\n *
Calls the specified function [block] with `this` value as its receiver and returns its encapsulated result if invocation
was successful,\n * catching any [Throwable] exception that was thrown from the [block] function execution and
encapsulating it as a failure.\n */\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <T, R>
T.runCatching(block: T.() -> R): Result<R> {\n    return try {\n    Result.success(block())\n    } catch (e:
Throwable) {\n    Result.failure(e)\n    }\n}\n\n// -- extensions --\n\n/**\n * Returns the encapsulated value if this
instance represents [success][Result.isSuccess] or throws the encapsulated [Throwable] exception\n * if it is
[failure][Result.isFailure].\n */\n * This function is a shorthand for `getOrElse { throw it }` (see [getOrElse]).\n
*/\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <T> Result<T>.getOrThrow(): T {\n
throwOnFailure()\n    return value as T\n}\n\n/**\n * Returns the encapsulated value if this instance represents
[success][Result.isSuccess] or the\n * result of [onFailure] function for the encapsulated [Throwable] exception
if it is [failure][Result.isFailure].\n */\n * Note, that this function rethrows any [Throwable] exception thrown by
[onFailure] function.\n */\n * This function is a shorthand for `fold(onSuccess = { it }, onFailure = onFailure)` (see
[fold]).\n */\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T : R> Result<T>.getOrElse(onFailure:
(exception: Throwable) -> R): R {\n    contract {\n        callsInPlace(onFailure,
InvocationKind.AT_MOST_ONCE)\n    }\n    return when (val exception = exceptionOrNull()) {\n        null ->
value as T\n        else -> onFailure(exception)\n    }\n}\n\n/**\n * Returns the encapsulated value if this instance
represents [success][Result.isSuccess] or the\n * [defaultValue] if it is [failure][Result.isFailure].\n */\n * This
function is a shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n
*/\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T : R> Result<T>.getOrDefault(defaultValue: R):
R {\n    if (isFailure) return defaultValue\n
return value as T\n}\n\n/**\n * Returns the result of [onSuccess] for the encapsulated value if this instance
represents [success][Result.isSuccess]\n * or the result of [onFailure] function for the encapsulated [Throwable]
exception if it is [failure][Result.isFailure].\n */\n * Note, that this function rethrows any [Throwable] exception
thrown by [onSuccess] or by [onFailure] function.\n */\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun
<R, T> Result<T>.fold(\n    onSuccess: (value: T) -> R,\n    onFailure: (exception: Throwable) -> R\n): R {\n
contract {\n        callsInPlace(onSuccess, InvocationKind.AT_MOST_ONCE)\n        callsInPlace(onFailure,
InvocationKind.AT_MOST_ONCE)\n    }\n    return when (val exception = exceptionOrNull()) {\n        null ->
onSuccess(value as T)\n        else -> onFailure(exception)\n    }\n}\n\n// transformation\n\n/**\n * Returns the
encapsulated result of the given [transform] function applied to the encapsulated value\n * if this
instance represents [success][Result.isSuccess] or the\n * original encapsulated [Throwable] exception if it is
[failure][Result.isFailure].\n */\n * Note, that this function rethrows any [Throwable] exception thrown by

```

[transform] function.
 * See [mapCatching] for an alternative that encapsulates exceptions.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.map(transform: (value: T) -> R):
Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    }
    return when {
        isSuccess -> Result.success(transform(value as T))
        else -> Result(value)
    }
}

```

* Returns the encapsulated result of the given [transform] function applied to the encapsulated value
 * if this instance represents [success][Result.isSuccess] or the original encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
 * See [map] for an alternative that rethrows exceptions from `transform` function.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R> {
    return when {
        isSuccess -> runCatching { transform(value as T) }
        else -> Result(value)
    }
}

```

* Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception
 * if this instance represents [failure][Result.isFailure] or the original encapsulated value if it is [success][Result.isSuccess].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.
 * See [recoverCatching] for an alternative that encapsulates exceptions.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    }
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> Result.success(transform(exception))
    }
}

```

* Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception
 * if this instance represents [failure][Result.isFailure] or the original encapsulated value if it is [success][Result.isSuccess].
 * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
 * See [recover] for an alternative that rethrows exceptions.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.recoverCatching(transform: (exception: Throwable) -> R): Result<R> {
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> runCatching { transform(exception) }
    }
}

```

* Performs the given [action] on the encapsulated [Throwable] exception if this instance represents [failure][Result.isFailure].
 * Returns the original `Result` unchanged.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    }
    exceptionOrNull()?.let { action(it) }
    return this
}

```

* Performs the given [action] on the encapsulated value if this instance represents [success][Result.isSuccess].
 * Returns the original `Result` unchanged.

```

*^@InlineOnly^@SinceKotlin("1.3")
public inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    }
    if (isSuccess) action(value as T)
    return this
}

```

 --\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

package kotlin.coroutines
import kotlin.contracts.*
import kotlin.coroutines.intrinsics.*
import kotlin.internal.InlineOnly

/**
 * Interface representing a continuation after a suspension point that returns a value of type `T`.
 */
@SinceKotlin("1.3")
public interface Continuation<in T> {
    /**
     * The context of the coroutine that corresponds to this continuation.
     */
    public val context: CoroutineContext

    /**
     * Resumes the execution of the corresponding coroutine passing a successful or failed [result] as the return value of the last suspension point.
     */
    public fun resumeWith(result: Result<T>)
}

/**
 * Classes and interfaces marked with this annotation are restricted when used as receivers for extension `suspend` functions. These `suspend` extensions can only invoke other member or extension `suspend` functions on this particular receiver and are restricted from calling arbitrary suspension functions.
 */
@SinceKotlin("1.3")
@Target(AnnotationTarget.CLASS)
@Retention(AnnotationRetention.BINARY)

```

```

blic annotation class RestrictsSuspension\n\n/**\n * Resumes the execution of the corresponding coroutine passing
[value] as the return value of the last suspension point.\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic inline
fun <T> Continuation<T>.resume(value: T): Unit =\n    resumeWith(Result.success(value))\n\n/**\n * Resumes the
execution of the corresponding coroutine so that the [exception] is re-thrown right after the\n * last suspension
point.\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic inline fun <T>
Continuation<T>.resumeWithException(exception: Throwable): Unit =\n
resumeWith(Result.failure(exception))\n\n\n/**\n * Creates a [Continuation] instance with the given [context] and
implementation of [resumeWith] method.\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic inline fun <T>
Continuation(\n    context: CoroutineContext,\n    crossinline resumeWith: (Result<T>) -> Unit)\n):
Continuation<T> =\n    object : Continuation<T> {\n        override val context: CoroutineContext\n            get() =
context\n        override fun resumeWith(result: Result<T>) =\n            resumeWith(result)\n    }\n\n/**\n * Creates
a coroutine without a receiver and with result type [T].\n * This function creates a new, fresh instance of
suspendable computation every time it is invoked.\n * To start executing the created coroutine, invoke
`resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when the
coroutine completes with a result or an exception.\n * Subsequent invocation of any resume function on the
resulting continuation will produce an [IllegalStateException].\n
*/\n@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <T> (suspend () ->
T).createCoroutine(\n    completion: Continuation<T>)\n): Continuation<Unit>
=\n    SafeContinuation(createCoroutineUnintercepted(completion).intercepted(),
COROUTINE_SUSPENDED)\n\n\n/**\n * Creates a coroutine with receiver type [R] and result type [T].\n * This
function creates a new, fresh instance of suspendable computation every time it is invoked.\n * To start
executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.\n * The
[completion] continuation is invoked when the coroutine completes with a result or an exception.\n * Subsequent
invocation of any resume function on the resulting continuation will produce an [IllegalStateException].\n
*/\n@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <R, T> (suspend R.() ->
T).createCoroutine(\n    receiver: R,\n    completion: Continuation<T>)\n): Continuation<Unit> =\n    SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(),
COROUTINE_SUSPENDED)\n\n\n/**\n * Starts a coroutine without a receiver and with result type [T].\n * This
function creates and starts a new, fresh instance of suspendable computation every time it is invoked.\n * The
[completion] continuation is invoked when the coroutine completes with a result or an exception.\n
*/\n@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <T> (suspend () ->
T).startCoroutine(\n    completion: Continuation<T>)\n) {\n
createCoroutineUnintercepted(completion).intercepted().resume(Unit)\n}\n\n\n/**\n * Starts a coroutine with receiver
type [R] and result type [T].\n * This function creates and starts a new, fresh instance of suspendable computation
every time it is invoked.\n * The [completion] continuation is invoked when the coroutine completes with a result
or an exception.\n */\n@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <R, T> (suspend
R.() -> T).startCoroutine(\n    receiver: R,\n    completion: Continuation<T>)\n) {\n
createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)\n}\n\n\n/**\n * Obtains
the current continuation instance inside suspend functions and suspends\n * the currently running coroutine.\n */\n
In this function both [Continuation.resume] and [Continuation.resumeWithException] can be used either
synchronously in\n * the same stack-frame where the suspension function is run or asynchronously later in the same
thread or\n * from a different thread of execution. Subsequent invocation of any resume function will produce an
[IllegalStateException].\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic suspend inline fun <T>
suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T {\n    contract { callsInPlace(block,
InvocationKind.EXACTLY_ONCE) }\n    return suspendCoroutineUninterceptedOrReturn { c: Continuation<T> -
->\n        val safe = SafeContinuation(c.intercepted())\n        block(safe)\n        safe.getOrThrow()\n    }\n}\n\n\n/**\n * Returns the context of the current coroutine.\n
*/\n@SinceKotlin("1.3")\n@Suppress("WRONG_MODIFIER_TARGET")\n@InlineOnly\npublic

```

```

suspend inline val coroutineContext: CoroutineContext\n    get() {\n        throw
NotImplementedError("Implemented as intrinsic")\n    }\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines.intrinsics\n\nimport
kotlin.coroutines.*\nimport kotlin.internal.InlineOnly\n\n/**\n * Starts an unintercepted coroutine without a receiver
and with result type [T] and executes it until its first suspension.\n * Returns the result of the coroutine or throws its
exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n * In the latter case, the
[completion] continuation is invoked when the coroutine completes with a result or an exception.\n */\n * The
coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might\n
* be present in
the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation\n * context
is established.\n */\n * This function is designed to be used from inside of
[suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference
to the suspending function.\n */\n\n@SinceKotlin("1.3")\n@InlineOnly\npublic actual inline fun <T> (suspend () ->
T).startCoroutineUninterceptedOrReturn(\n    completion: Continuation<T>)\n): Any? =
this.asDynamic()(completion, false)\n\n/**\n * Starts an unintercepted coroutine with receiver type [R] and result
type [T] and executes it until its first suspension.\n * Returns the result of the coroutine or throws its exception if it
does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n * In the latter case, the [completion]
continuation is invoked when the coroutine completes with a result or an exception.\n */\n * The coroutine is started
directly in the invoker's thread without
going through the [ContinuationInterceptor] that might\n * be present in the completion's [CoroutineContext]. It is
the invoker's responsibility to ensure that a proper invocation\n * context is established.\n */\n * This function is
designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the
suspended\n * coroutine using a reference to the suspending function.\n\n*/\n\n@SinceKotlin("1.3")\n@InlineOnly\npublic actual inline fun <R, T> (suspend R.() ->
T).startCoroutineUninterceptedOrReturn(\n    receiver: R,\n    completion: Continuation<T>)\n): Any? =
this.asDynamic()(receiver, completion, false)\n\n@InlineOnly\ninternal actual inline fun <R, P, T> (suspend R.(P) -
> T).startCoroutineUninterceptedOrReturn(\n    receiver: R,\n    param: P,\n    completion: Continuation<T>)\n):
Any? = this.asDynamic()(receiver, param, completion, false)\n\n/**\n * Creates unintercepted coroutine without
receiver and with result type [T].\n * This function creates
a new, fresh instance of suspendable computation every time it is invoked.\n */\n * To start executing the created
coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is
invoked when coroutine completes with result or exception.\n */\n * This function returns unintercepted
continuation.\n * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going
through the\n * [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the
invoker's responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this
function may get invoked in an arbitrary context.\n */\n * [Continuation.intercepted] can be used to acquire the
intercepted continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n
* both the coroutine and [completion] happens in the invocation context established
by\n * [ContinuationInterceptor].\n */\n * Repeated invocation of any resume function on the resulting continuation
corrupts the\n * state machine of the coroutine and may result in arbitrary behaviour or exception.\n\n*/\n\n@SinceKotlin("1.3")\npublic actual fun <T> (suspend () -> T).createCoroutineUnintercepted(\n    completion:
Continuation<T>)\n): Continuation<Unit> =\n    // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n    if (this.asDynamic().length == 2) {\n        // When `suspended` is true the continuation is created, but not
executed\n        this.asDynamic()(completion, true)\n    } else {\n        createCoroutineFromSuspendFunction(completion) {\n            this.asDynamic()(completion)\n        }\n    }\n\n/**\n * Creates unintercepted coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh
instance of suspendable computation every time it is invoked.\n */\n * To start executing the created coroutine,

```

invoke `resume(Unit)`

on the returned [Continuation] instance.\n * The [completion] continuation is invoked when coroutine completes with result or exception.\n * This function returns unintercepted continuation.\n * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n * [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function may get invoked in an arbitrary context.\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n * Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of

the coroutine and may result in arbitrary behaviour or exception.\n *
@SinceKotlin("1.3")\npublic actual fun <R, T> (suspend R.() -> T).createCoroutineUnintercepted(\n receiver: R,\n completion: Continuation<T>)\n: Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n if (this.asDynamic().length == 3) {\n // When `suspended` is true the continuation is created, but not executed\n this.asDynamic()(receiver, completion, true)\n } else {\n createCoroutineFromSuspendFunction(completion)\n this.asDynamic()(receiver, completion)\n }\n\n\n * Intercepts this continuation with [ContinuationInterceptor].\n * This function shall be used on the immediate result of [createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn],\n * in which case it checks for [ContinuationInterceptor] in the continuation's [context][Continuation.context],\n * invokes [ContinuationInterceptor.interceptContinuation],

caches and returns the result.\n * If this function is invoked on other [Continuation] instances it returns `this` continuation unchanged.\n *
@SinceKotlin("1.3")\npublic actual fun <T> Continuation<T>.intercepted():

Continuation<T> =\n (this as? CoroutineImpl)?.intercepted() ?: this\n\nprivate inline fun <T> createCoroutineFromSuspendFunction(\n completion: Continuation<T>,\n crossinline block: () -> Any?)\n: Continuation<Unit> {\n @Suppress("UNCHECKED_CAST")\n return object : CoroutineImpl(completion as Continuation<Any?>) {\n override fun doResume(): Any? {\n exception?.let { throw it }\n return block()\n }\n }\n}\n\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.js\n// Mirrors signature from JS IR BE\n// Used for js.translator/testData/box/number/mulInt32.kt\n@library\n@jsName("imulEmulated")\n@Suppress("UNUSED_PARAMETER")\ninternal fun imul(x: Int, y: Int): Int =

definedExternally\n\n@Suppress("NOTHING_TO_INLINE")\ninternal inline fun isArrayish(o: dynamic) = js("Kotlin").isArrayish(o)\n\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin\n// NOTE: Do not author your exceptions as they are written in this file, instead use this template:\n\npublic open class MyException : Exception {\n constructor() : super()\n constructor(message: String?) : super(message)\n constructor(message: String?, cause: Throwable?) : super(message, cause)\n constructor(cause: Throwable?) : super(cause)\n}\n\n\n// TODO: remove primary constructors, make all secondary\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic actual open class Error actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {\n actual constructor() : this(null, null)\n actual constructor(message: String?) : this(message, null)\n actual constructor(cause: Throwable?) : this(undefiend, cause)\n}\n\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic actual open class Exception actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {\n actual constructor() : this(null, null)\n actual constructor(message: String?) : this(message, null)\n actual constructor(cause: Throwable?) : this(undefiend, cause)\n}\n\n\npublic actual open class RuntimeException actual constructor(message: String?, cause: Throwable?) : Exception(message, cause) {\n actual constructor() : this(null, null)\n actual constructor(message: String?) : this(message, null)\n actual constructor(cause: Throwable?) : this(undefiend,

```

cause)\n}\n\npublic actual open
class IllegalArgumentException actual constructor(message: String?, cause: Throwable?) :
RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual constructor(message:
String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefiend, cause)\n}\n\npublic actual
open class IllegalStateException actual constructor(message: String?, cause: Throwable?) :
RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual constructor(message:
String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefiend, cause)\n}\n\npublic actual
open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message) {\n
actual constructor() : this(null)\n}\n\npublic actual open class ConcurrentModificationException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual
constructor(message: String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefiend,
cause)\n}\n\npublic actual open class UnsupportedOperationException actual constructor(message: String?, cause:
Throwable?) : RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual
constructor(message: String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefiend,
cause)\n}\n\npublic actual open class NumberFormatException actual constructor(message: String?) :
IllegalArgumentException(message) {\n  actual constructor() : this(null)\n}\n\npublic actual open class
NullPointerException actual constructor(message: String?) : RuntimeException(message) {\n  actual constructor() :
this(null)\n}\n\npublic actual open class ClassCastException actual constructor(message: String?) :
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\npublic actual open class
AssertionError\n@SinceKotlin("1.4")\nconstructor(message:
String?, cause: Throwable?) : Error(message, cause) {\n  actual constructor() : this(null)\n  constructor(message:
String?) : this(message, null)\n  actual constructor(message: Any?) : this(message.toString(), message as?
Throwable)\n}\n\npublic actual open class NoSuchElementException actual constructor(message: String?) :
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\n@SinceKotlin("1.3")\npublic actual open
class ArithmeticException actual constructor(message: String?) : RuntimeException(message) {\n  actual
constructor() : this(null)\n}\n\npublic actual open class NoWhenBranchMatchedException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual constructor(cause:
Throwable?) : this(undefiend, cause)\n}\n\npublic actual open class UninitializedPropertyAccessException
actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual
constructor() : this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual
constructor(cause: Throwable?) : this(undefiend, cause)\n}\n\n", /*\n * Copyright 2010-2019 JetBrains s.r.o. Use of
this source code is governed by the Apache 2.0 license\n * that can be found in the license/LICENSE.txt file.\n
*\n\n@file:Suppress("UNUSED_PARAMETER")\n\npackage kotlin.js\n\n@kotlin.internal.InlineOnly\n\ninternal
inline fun jsDeleteProperty(obj: Any, property: Any) {\n  js("delete
obj[property]")\n}\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =\n
js("lhs | rhs").unsafeCast<Int>()", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage
kotlin.math\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n * \n * If [sign] is `NaN` the
sign of the result is undefined.\n */\n@SinceKotlin("1.2")\npublic actual fun Double.withSign(sign: Double):
Double {\n  val thisSignBit = js("Kotlin").doubleSignBit(this).unsafeCast<Int>()\n  val newSignBit =
js("Kotlin").doubleSignBit(sign).unsafeCast<Int>()\n  return if (thisSignBit == newSignBit) this else -
this\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n *
according to the IEEE 754 floating-point "double format" bit layout.\n

```

```

*^@SinceKotlin("1.2")^@library("doubleToBits")^public actual fun Double.toBits(): Long =
definedExternally^/^/**^ * Returns a bit representation
of the specified floating-point value as [Long]^ * according to the IEEE 754 floating-point "double format" bit
layout,^ * preserving `NaN` values exact layout.^
*^@SinceKotlin("1.2")^@library("doubleToRawBits")^public actual fun Double.toRawBits(): Long =
definedExternally^/^/**^ * Returns the [Double] value corresponding to a given bit representation.^
*^@SinceKotlin("1.2")^@kotlin.internal.InlineOnly^public actual inline fun Double.Companion.fromBits(bits:
Long): Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()^/^/**^ * Returns a bit representation
of the specified floating-point value as [Int]^ * according to the IEEE 754 floating-point "single format" bit
layout.^ * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,^ * so
some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.^
*^@SinceKotlin("1.2")^@library("floatToBits")^public
actual fun Float.toBits(): Int = definedExternally^/^/**^ * Returns a bit representation of the specified floating-
point value as [Int]^ * according to the IEEE 754 floating-point "single format" bit layout,^ * preserving `NaN`
values exact layout.^ * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can
represent,^ * so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and
back.^
*^@SinceKotlin("1.2")^@library("floatToRawBits")^public actual fun Float.toRawBits(): Int =
definedExternally^/^/**^ * Returns the [Float] value corresponding to a given bit representation.^
*^@SinceKotlin("1.2")^@kotlin.internal.InlineOnly^public actual inline fun Float.Companion.fromBits(bits:
Int): Float =
js("Kotlin").floatFromBits(bits).unsafeCast<Float>()^/^@Suppress("NOTHING_TO_INLINE")^internal
inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()^internal
inline val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()^internal inline val Long.high: Int
get() = this.asDynamic().getHighBits().unsafeCast<Int>()^/^/**^ * Copyright 2010-2020 JetBrains s.r.o. and
Kotlin Programming Language contributors.^ * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.^
*^@nimport kotlin.reflect.KClass^@PublishedApi^internal fun
<T : Annotation> KClass<*>.findAssociatedObject(@Suppress("UNUSED_PARAMETER") annotationClass:
KClass<T>): Any? {^ * // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir.^
return null^}^/^/**^ * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.^ *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.^
*^@npackage kotlin.text^/^/**^ * Returns a string representation of this [Long] value
in the specified [radix].^ * @throws IllegalArgumentException when [radix] is not a valid radix for number to
string conversion.^
*^@SinceKotlin("1.2")^public actual fun Long.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))^/^/**^ * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.^ * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.^
*^@npackage kotlin.text^/^@NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt^/^ See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib^/^@n^/^ 1343 ranges
totally^private object Category {^ * val decodedRangeStart: IntArray^ * val decodedRangeCategory: IntArray^
^ * init {^ * val toBase64 =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"^ * val fromBase64 =
IntArray(128)^ * for (i in toBase64.indices) {^ * fromBase64[toBase64[i].code]
= i^ * }^ * }^ * // rangeStartDiff.length = 1482^ * val rangeStartDiff =
"gBCFEDCKCDCaDDaDBhBCEEDDDDDDEDXBHYBH5BRwBGDCHDCIDFHDFHDCDEIRTEE7BGHDDJI
CBbSEMOFGERwDEDDDDDECEFCRBjBhBFDcYFFCCzBvBjBBFC3BohDBmBDGpBDDCtBBJlBEECLGDFC
LDCgBBKVKEDiDDHCFECECKCEODBebC5CLBOKhBJDDDDWEBHFcfCPBZDEL1BVBSLPBgBB2BDB
DICFBHKCKCPDBHEDWBHEDDDDEDEDIBDGDCCKCGDDDCGECCWBFMDDCEDDDCHDDHKDDBK
DBHFCWBFgFDBDDFEDBPDDKCHBGDCHEDWBFgFDCEDEDBHDDGDCKCGJEGDBFDDFDDDDDME
FDBFDCGBOKDFDFDCGFCXBQDDDDDBEGEDFDDKHBHDDGFCXBKBFCEFCFCHECCKDNCCHFC

```



```

0x1f\n    }\n }\n}\n\n/**\n * Returns the Unicode general category of this character as an Int.\n *\ninternal
fun Char.getCategoryValue(): Int {\n    val ch = this.code\n    val index =
binarySearchRange(Category.decodedRangeStart, ch)\n    val start = Category.decodedRangeStart[index]\n    val
code = Category.decodedRangeCategory[index]\n    val value = categoryValueFrom(code, ch - start)\n\n    return if
(value == 17) CharCategory.UNASSIGNED.value else value\n}\n\ninternal fun decodeVarLenBase64(base64:
String, fromBase64: IntArray, resultLength: Int): IntArray {\n    val result = IntArray(resultLength)\n
    var index = 0\n    var int = 0\n    var shift = 0\n    for (char in base64) {\n        val sixBit =
fromBase64[char.code]\n        int = int or ((sixBit and 0x1f) shl shift)\n        if (sixBit < 0x20) {\n
result[index++] = int\n            int = 0\n            shift = 0\n        } else {\n            shift += 5\n        }\n    }\n    return
result\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport
kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Reverses elements in the list in-
place.\n *\npublic actual fun <T> MutableList<T>.reverse(): Unit {\n    val midPoint
= (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n
val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n
}\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n//
See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\n// 37 ranges totally\nprivate object Digit
{\n    internal val rangeStart = intArrayOf(\n        0x0030, 0x0660, 0x06f0, 0x07c0, 0x0966, 0x09e6, 0x0a66,
0x0ae6, 0x0b66, 0x0be6, 0x0c66, 0x0ce6, 0x0d66, 0x0de6, 0x0e50, 0x0ed0, 0x0f20, 0x1040, 0x1090, 0x17e0, \n
0x1810, 0x1946, 0x19d0, 0x1a80, 0x1a90, 0x1b50, 0x1bb0, 0x1c40, 0x1c50, 0xa620,
0xa8d0, 0xa900, 0xa9d0, 0xa9f0, 0xaa50, 0xabf0, 0xff10, \n    )\n}\n\n/**\n * Returns the index of the largest
element in [array] smaller or equal to the specified [needle],\n * or -1 if [needle] is smaller than the smallest element
in [array].\n *\ninternal fun binarySearchRange(array: IntArray, needle: Int): Int {\n    var bottom = 0\n    var top =
array.size - 1\n    var middle = -1\n    var value = 0\n    while (bottom <= top) {\n        middle = (bottom + top) / 2\n
value = array[middle]\n        if (needle > value)\n            bottom = middle + 1\n        else if (needle == value)\n
return middle\n        else\n            top = middle - 1\n    }\n    return middle - (if (needle < value) 1 else
0)\n}\n\n/**\n * Returns an integer from 0..9 indicating the digit this character represents,\n * or -1 if this character
is not a digit.\n *\ninternal fun Char.digitToIntImpl(): Int {\n    val ch = this.code\n    val index =
binarySearchRange(Digit.rangeStart, ch)\n
    val diff = ch - Digit.rangeStart[index]\n    return if (diff < 10) diff else -1\n}\n\n/**\n * Returns `true` if this
character is a digit.\n *\ninternal fun Char.isDigitImpl(): Boolean {\n    return digitToIntImpl() >= 0\n}\n\n", "/*\n *
Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\npackage
kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\n// 222 ranges totally\nprivate object Letter {\n
val decodedRangeStart: IntArray\n    val decodedRangeLength: IntArray\n    val decodedRangeCategory: IntArray\n
\n    init {\n        val toBase64 =
\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/\n        val fromBase64 =
IntArray(128)\n        for (i in toBase64.indices) {\n            fromBase64[toBase64[i].code]
= i\n        }\n        \n        // rangeStartDiff.length = 356\n        val rangeStartDiff =
\"hCgBpCQGYHZH5BRpBPPPPPRMP5BPPICPP6BkEPPPPcXPzBvBrB3BOiDoBHwD+E3DauCnFmBmB2D
6E1BIBTiBmBIBP5BhBiBrBvBjBqBnBPRtBiCmCtBIB0BmB5BiB7BmBgEmChBzGCoEoGVpBsFrhBPqKQ2B
wBYoFgB4CJuTiEvBuCuDrF5DgEgFIJ1DgFmBQtBsBRGsB+BPiBID1E1jDPRPPPPPPPPPGQSQS/DxENVNU+
B9zCwBwBPPCkDPNnBPqDYY1R8B7FkFgTgwGgwUwmBgKwBuBScmEP/BPPPPPrBP8B7F1B/ErBqC6B7B

```



```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.asList(): List<UInt> {\n
return object : AbstractList<UInt>(),
    RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean =
this@asList.isEmpty()\n        override fun contains(element: UInt): Boolean = this@asList.contains(element)\n
override fun get(index: Int): UInt {\n            AbstractList.checkElementIndex(index, size)\n            return
this@asList[index]\n        }\n        override fun indexOf(element: UInt): Int {\n
@Suppress("USELESS_CAST")\n            if ((element as Any?) !is UInt) return -1\n            return
this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element: UInt): Int {\n
@Suppress("USELESS_CAST")\n            if ((element as Any?) !is UInt) return -1\n            return
this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.asList(): List<ULong>
{\n    return
object : AbstractList<ULong>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n
override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: ULong): Boolean =
this@asList.contains(element)\n        override fun get(index: Int): ULong {\n
AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun
indexOf(element: ULong): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is
ULong) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element:
ULong): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is ULong) return -1\n
return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.asList():
List<UByte> {\n    return object : AbstractList<UByte>(), RandomAccess {\n        override val size: Int get() =
this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun
contains(element: UByte): Boolean = this@asList.contains(element)\n        override fun get(index: Int): UByte {\n
AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun
indexOf(element: UByte): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is
UByte) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element:
UByte): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is UByte) return -1\n
return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
actual fun UShortArray.asList(): List<UShort> {\n    return object : AbstractList<UShort>(), RandomAccess {\n
override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n
override fun contains(element: UShort): Boolean = this@asList.contains(element)\n        override fun get(index:
Int): UShort {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n
override fun indexOf(element: UShort): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as
Any?) !is UShort) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun
lastIndexOf(element: UShort): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is
UShort) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n", /*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\npackage kotlin.text\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n// 9 ranges
totally\n/**\n * Returns `true` if this character is a whitespace.\n
*\ninternal fun Char.isWhitespaceImpl(): Boolean
{\n    val ch = this.code\n    return ch in 0x0009..0x000d\n        || ch in 0x001c..0x0020\n        || ch == 0x00a0\n
        || ch > 0x1000 && (\n            ch == 0x1680\n            || ch in 0x2000..0x200a\n            || ch == 0x2028\n
            || ch == 0x2029\n            || ch == 0x202f\n            || ch == 0x205f\n            || ch == 0x3000\n
        )\n}\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache

```

2.0 license that can be found in the license/LICENSE.txt file.

```

package kotlin
public actual fun interface
Comparator<T> {
    @JsName("compare")
    public actual fun compare(a: T, b: T): Int
}

```

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

package kotlin.js
import
kotlin.annotation.AnnotationTarget.*
@Target(FUNCTION)
@Deprecated("Use inline extension function with
body using dynamic")
public annotation class nativeGetter
@Target(FUNCTION)
@Deprecated("Use inline
extension function with body using dynamic")
public annotation class
nativeSetter
@Target(FUNCTION)
@Deprecated("Use inline extension function with body using
dynamic")
public annotation class nativeInvoke
@Target(CLASS, FUNCTION, PROPERTY)
internal
annotation class library(public val name: String = "")
@Target(CLASS)
internal
annotation class marker/**
 * Gives a declaration (a function, a property or a class) specific name in
JavaScript.
 * This may be useful in the following cases:
 * * There are two functions for which the
compiler gives same name in JavaScript, you can
 * mark one with `@JsName(...)` to prevent the compiler from
reporting error.
 * * You are writing a JavaScript library in Kotlin. The compiler produces mangled names
 * for functions with parameters, which is unnatural for usual JavaScript developer.
 * You can put `@JsName(...)`
on functions you want to be available from JavaScript.
 * * For some reason you want to rename declaration, e.g.
there's common term in JavaScript
 * for a concept provided by the declaration, which is uncommon in Kotlin.
 * Example:
 * ```
kotlin
class Person(val name: String) {
    fun hello() {
        println("Hello
$name!")
    }
    @JsName("helloWithGreeting")
    fun hello(greeting: String) {
        println("$greeting $name!")
    }
}
 * @property
name the name which compiler uses both for declaration itself and for all references to the declaration.
 * It's
required to denote a valid JavaScript identifier.
 */
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR,
PROPERTY_GETTER, PROPERTY_SETTER)
public actual annotation class JsName(actual val name:
String)
/**
 * Denotes an `external` declaration that must be imported from native JavaScript library.
 * The compiler produces the code relevant for the target module system, for example, in case of CommonJS,
 * it will import the declaration via the `require(...)` function.
 * The annotation can be used on top-level external
declarations (classes, properties, functions) and files.
 * In case of file (which can't be `external`) the following rule
applies: all the declarations in
 * the file must be `external`. By applying `@JsModule(...)` on a file you tell the compiler to import a JavaScript
object
 * that contain all the declarations from the file.
 * Example:
 * ```
kotlin
@JsModule("jquery")
external abstract class JQuery() {
    // some declarations here
}
 * @JsModule("jquery")
external fun JQuery(element: Element): JQuery
 * @property import name
of a module to import declaration from.
 * It is not interpreted by the Kotlin compiler, it's passed as is
directly to the target module system.
 * @see JsNonModule
 */
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public
annotation class JsModule(val import: String)
/**
 * Denotes an `external` declaration that can be used without
module system.
 * By default, an `external` declaration is available regardless your target module system.
 * However, by applying [JsModule] annotation you can make a
 * declaration unavailable to `plain` module system.
 * Some JavaScript libraries are distributed both as a standalone
downloadable piece of JavaScript and as a module available
 * as an npm package.
 * To tell the Kotlin compiler
to accept both cases, you can augment [JsModule] with the `@JsNonModule` annotation.
 * For example:
 * ```
kotlin
@JsModule("jquery")
@JsNonModule
@JsName("$")
external abstract class JQuery()
{
    // some declarations here
}
 * @JsModule("jquery")
@JsNonModule
@JsName("$")
external fun JQuery(element: Element): JQuery
 * @see JsModule
 */
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public
annotation class JsNonModule
/**
 * Adds prefix to `external` declarations in a source file.
 * JavaScript
does not have concept of packages (namespaces). They are usually emulated by nested objects.
 * The compiler

```

turns references to `external` declarations

either to plain unprefix names (in case of `*plain*` modules) or to plain imports. However, if a JavaScript library provides its declarations in packages, you won't be satisfied with this. You can tell the compiler to generate additional prefix before references to `external` declarations using the `@JsQualifier(...)` annotation. Note that a file marked with the `@JsQualifier(...)` annotation can't contain non-`external` declarations.

Example: `@file:JsQualifier("my.jsPackageName") package some.kotlinPackage external fun foo(x: Int) external fun bar(): String` @property value the qualifier to add to the declarations in the generated code. It must be a sequence of valid JavaScript identifiers separated by the `.` character. Examples of valid qualifiers are: `foo`, `bar.Baz`, `_.$.f`.

`@Retention(AnnotationRetention.BINARY) @Target(AnnotationTarget.FILE) public annotation class JsQualifier(val value: String)` Exports top-level declaration on JS platform. Compiled module exposes declarations that are marked with this annotation without name mangling. This annotation can be applied to either files or top-level declarations. It is currently prohibited to export the following kinds of declarations: `* expect` declarations, inline functions with reified type parameters, suspend functions, secondary constructors without `@JsName` extension properties, enum classes, annotation classes. Signatures of exported declarations must only contain "exportable" types: `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`, `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`, `Array<exportable-type>`. Function types with exportable parameters and return types.

`* external` or `@JsExport` classes and interfaces. Nullable counterparts of types above. Unit return type. Must not be nullable. This annotation is experimental, meaning that restrictions mentioned above are subject to change. `@ExperimentalJsExport @Retention(AnnotationRetention.BINARY) @Target(CLASS, PROPERTY, FUNCTION, FILE) @SinceKotlin("1.3") public actual annotation class JsExport` Forces a top-level property to be initialized eagerly, opposed to lazily on the first access to file and/or property. `@ExperimentalStdlibApi @Retention(AnnotationRetention.BINARY) @Target(AnnotationTarget.PROPERTY) @SinceKotlin("1.6") @Deprecated("This annotation is a temporal migration assistance and may be removed in the future releases, please consider filing an issue about the case where it is needed") public annotation class EagerInitialization`, Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. `package kotlin.jvm` these are used in common generated code in `stdlib`. TODO: find how to deprecate these ones. `@Target(AnnotationTarget.FIELD) @Retention(AnnotationRetention.SOURCE) public actual annotation class Volatile @Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER,`

`AnnotationTarget.PROPERTY_SETTER) @Retention(AnnotationRetention.SOURCE) public actual annotation class Synchronized`, Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. `package kotlin.collections` Provides a skeletal implementation of the `[MutableCollection]` interface. @param E the type of elements contained in the collection. The collection is invariant in its element type.

```
* public actual abstract class AbstractMutableCollection<E> protected actual constructor() :
AbstractCollection<E>(), MutableCollection<E> {
    actual abstract override fun add(element: E): Boolean
    actual override fun remove(element: E): Boolean {
        checkIsMutable()
        val iterator = iterator()
        while (iterator.hasNext()) {
            if (iterator.next() == element) {
                iterator.remove()
                return true
            }
        }
        return false
    }
    actual override fun addAll(elements: Collection<E>): Boolean {
        checkIsMutable()
        var modified = false
        for (element in elements) {
            if (add(element))
                modified = true
        }
        return modified
    }
    actual override fun removeAll(elements: Collection<E>): Boolean {
        checkIsMutable()
        return (this as MutableIterable<E>).removeAll { it in elements }
    }
}
```



```

get(last)\n    }\n\n    override fun remove() {\n        check(last != -1) { \"Call next() or previous() before
removing element from the iterator.\" }\n        removeAt(last)\n        index = last\n        last = -1\n    }\n\n    /**\n     * Implementation of `MutableListIterator` for abstract lists.\n     */\n    private inner class
ListIteratorImpl(index: Int) : IteratorImpl(), MutableListIterator<E> {\n        init {\n            AbstractList.checkPositionIndex(index, this@AbstractMutableList.size)\n            this.index = index\n        }\n\n        override fun hasPrevious(): Boolean = index > 0\n        override fun nextIndex(): Int = index\n        override fun
previous(): E {\n            if (!hasPrevious()) throw NoSuchElementException()\n            last = --index\n            return get(last)\n        }\n\n        override fun previousIndex(): Int = index - 1\n        override fun add(element: E) {\n            add(index,
element)\n            index++\n            last = -1\n        }\n        override fun set(element: E) {\n            check(last != -
1) { \"Call next() or previous() before updating element value with the iterator.\" }\n            set(last, element)\n
        }\n    }\n\n    private class SubList<E>(private val list: AbstractMutableList<E>, private val fromIndex: Int, toIndex:
Int) : AbstractMutableList<E>(), RandomAccess {\n        private var _size: Int = 0\n        init {\n            AbstractList.checkRangeIndexes(fromIndex, toIndex, list.size)\n            this._size = toIndex - fromIndex\n        }\n\n        override fun add(index: Int, element: E) {\n            AbstractList.checkPositionIndex(index, _size)\n            list.add(fromIndex
+ index, element)\n            _size++\n        }\n        override fun get(index: Int): E {\n            AbstractList.checkElementIndex(index, _size)\n            return list[fromIndex + index]\n        }\n        override
fun removeAt(index: Int): E {\n            AbstractList.checkElementIndex(index, _size)\n            val result =
list.removeAt(fromIndex + index)\n            _size--\n            return result\n        }\n        override fun set(index: Int,
element: E): E {\n            AbstractList.checkElementIndex(index, _size)\n            return list.set(fromIndex + index,
element)\n        }\n        override val size: Int get() = _size\n        internal override fun checkIsMutable(): Unit =
list.checkIsMutable()\n    }\n\n}\n\n\"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n\n * Based
on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Provides a
skeletal implementation of the [MutableMap] interface.\n */\n * The implementor is required to implement [entries]
property, which should return mutable set of map entries, and [put] function.\n */\n * @param K the type of map
keys. The map is invariant in its key type.\n * @param V the type of map values. The map is invariant in its value
type.\n */\n\npublic actual abstract class AbstractMutableMap<K, V> protected actual constructor() : AbstractMap<K,
V>(), MutableMap<K, V> {\n    /**\n     * A mutable [Map.Entry] shared by several [Map] implementations.\n     */\n    internal open class SimpleEntry<K, V>(override val key: K, value: V) : MutableMap.MutableEntry<K, V>
{\n        constructor(entry: Map.Entry<K, V>) : this(entry.key, entry.value)\n        private var _value = value\n\n        override val value: V get() = _value\n        override fun setValue(newValue: V): V {\n
            // Should check if the map containing this entry is mutable.\n            // However, to not increase entry
memory footprint it might be worthwhile not to check it here and\n            // force subclasses that implement
`build()` (freezing) operation to implement their own `MutableEntry`.\n\n            this@AbstractMutableMap.checkIsMutable()\n            val oldValue = this._value\n            this._value = newValue\n\n            return oldValue\n        }\n        override fun hashCode(): Int = entryHashCode(this)\n        override fun
toString(): String = entryToString(this)\n        override fun equals(other: Any?): Boolean = entryEquals(this,
other)\n    }\n\n    // intermediate abstract class to workaround KT-43321\n    internal abstract class
AbstractEntrySet<E : Map.Entry<K, V>, K, V> : AbstractMutableSet<E>() {\n        final override fun
contains(element: E): Boolean = containsEntry(element)\n        abstract fun containsEntry(element: Map.Entry<K,
V>): Boolean\n\n        final override fun remove(element: E): Boolean = removeEntry(element)\n        abstract fun
removeEntry(element: Map.Entry<K, V>): Boolean\n    }\n\n    actual override fun clear() {\n        entries.clear()\n
}\n\n    private var _keys: MutableSet<K>? = null\n    actual override val keys: MutableSet<K>\n        get() {\n            if (_keys == null) {\n                _keys = object : AbstractMutableSet<K>() {\n                    override fun

```

```

add(element: K): Boolean = throw UnsupportedOperationException("Add is not supported on keys")\n
override fun clear() {\n          this@AbstractMutableMap.clear()\n          }\n          override
operator fun contains(element: K): Boolean = containsKey(element)\n          override operator fun iterator():
MutableIterator<K> {\n          val entryIterator = entries.iterator()\n          return object :
MutableIterator<K> {\n
          override fun hasNext(): Boolean = entryIterator.hasNext()\n          override fun next(): K =
entryIterator.next().key\n          override fun remove() = entryIterator.remove()\n          }\n
          }\n          override fun remove(element: K): Boolean {\n          checkIsMutable()\n
if (containsKey(element)) {\n          this@AbstractMutableMap.remove(element)\n
return true\n          }\n          return false\n          }\n          override val size: Int get() =
this@AbstractMutableMap.size\n          override fun checkIsMutable(): Unit =
this@AbstractMutableMap.checkIsMutable()\n          }\n          }\n          return _keys!!\n          }\n          actual
abstract override fun put(key: K, value: V): V?\n          actual override fun putAll(from:
Map<out K, V>) {\n          checkIsMutable()\n          for ((key, value) in from) {\n          put(key, value)\n          }\n
}\n          private var _values: MutableCollection<V>? = null\n          actual override val values: MutableCollection<V>\n
get() {\n          if (_values == null) {\n          _values = object : AbstractMutableCollection<V>() {\n
          override fun add(element: V): Boolean = throw UnsupportedOperationException("Add is not supported on
values")\n          override fun clear() = this@AbstractMutableMap.clear()\n          override operator
fun contains(element: V): Boolean = containsValue(element)\n          override operator fun iterator():
MutableIterator<V> {\n          val entryIterator = entries.iterator()\n          return object :
MutableIterator<V> {\n
          override fun hasNext(): Boolean = entryIterator.hasNext()\n
          override fun
next(): V = entryIterator.next().value\n          override fun remove() = entryIterator.remove()\n
          }\n          }\n          }\n          override val size: Int get() = this@AbstractMutableMap.size\n
          override fun checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n          }\n          }\n
return _values!!\n          }\n          actual override fun remove(key: K): V? {\n          checkIsMutable()\n          val iter =
entries.iterator()\n          while (iter.hasNext()) {\n          val entry = iter.next()\n          val k = entry.key\n          if
(key == k) {\n          val value = entry.value\n          iter.remove()\n          return value\n          }\n
}\n          return null\n          }\n          }\n          /**\n          * This method is called every time when a mutating method is called on
this mutable map.\n          * Mutable maps that are built (frozen) must throw `UnsupportedOperationException`.\n
          */\n          internal open fun checkIsMutable(): Unit {\n          }\n          }\n          /**\n          * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n          * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n          */\n          package kotlin.collections\n          /**\n          * Provides a skeletal
implementation of the [MutableSet] interface.\n          */\n          @param E the type of elements contained in the set. The set is
invariant in its element type.\n          */\n          public actual abstract class AbstractMutableSet<E> protected actual constructor()
: AbstractMutableCollection<E>(), MutableSet<E> {\n          }\n          /**\n          * Compares this set with another set instance
with the unordered structural equality.\n          */\n          * @return `true`, if [other] instance is a [Set] of the same size, all
elements of which are contained in this set.\n          */\n          override fun equals(other: Any?): Boolean {\n          if (other
=== this) return true\n          if (other !is
Set<*>) return false\n          return AbstractSet.setEquals(this, other)\n          }\n          }\n          /**\n          * Returns the hash code
value for this set.\n          */\n          override fun hashCode(): Int = AbstractSet.unorderedHashCode(this)\n          }\n          }\n
          Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n          * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n          */\n          package
kotlin.collections\n          /**\n          * Provides a [MutableList] implementation, which uses a resizable array as its backing
storage.\n          */\n          * This implementation doesn't provide a way to manage capacity, as backing JS array is resizable
itself.\n          * There is no speed advantage to pre-allocating array sizes in JavaScript, so this implementation does not
include any of the\n          * capacity and "growth increment" concepts.\n          */\n          public actual open class ArrayList<E>
internal constructor(private var array: Array<Any?>) : AbstractMutableList<E>(), MutableList<E>,

```

```

RandomAccess {\n  private var isReadOnly: Boolean = false\n\n  /**\n   * Creates an empty [ArrayList].\n  *\n  public actual constructor() : this(emptyArray()) {\n\n  /**\n   * Creates an empty [ArrayList].\n   *\n  @param initialCapacity initial capacity (ignored)\n  *\n  public actual constructor(initialCapacity: Int) :\n  this(emptyArray()) {\n\n  /**\n   * Creates an [ArrayList] filled from the [elements] collection.\n   *\n  public\n  actual constructor(elements: Collection<E>) : this(elements.toArray<Any?>()) {\n\n  @PublishedApi\n  internal fun build(): List<E> {\n    checkIsMutable()\n    isReadOnly = true\n    return this\n  }\n\n  /**\n  Does nothing in this ArrayList implementation. *\n  public actual fun trimToSize() {\n\n  /**\n  Does nothing in\n  this ArrayList implementation. *\n  public actual fun ensureCapacity(minCapacity: Int) {\n\n  actual override\n  val size: Int get() = array.size\n  @Suppress("UNCHECKED_CAST")\n\n  actual override fun get(index: Int): E = array[rangeCheck(index)] as E\n  actual override fun set(index: Int,\n  element: E): E {\n    checkIsMutable()\n    rangeCheck(index)\n    @Suppress("UNCHECKED_CAST")\n    return array[index].apply { array[index] = element } as E\n  }\n\n  actual override fun add(element: E): Boolean\n  {\n    checkIsMutable()\n    array.asDynamic().push(element)\n    modCount++\n    return true\n  }\n\n  actual override fun add(index: Int, element: E): Unit {\n    checkIsMutable()\n    array.asDynamic().splice(insertionRangeCheck(index), 0, element)\n    modCount++\n  }\n\n  actual override\n  fun addAll(elements: Collection<E>): Boolean {\n    checkIsMutable()\n    if (elements.isEmpty()) return\n    false\n\n    array += elements.toArray<Any?>()\n    modCount++\n    return true\n  }\n\n  actual\n  override fun addAll(index: Int, elements: Collection<E>): Boolean {\n\n    checkIsMutable()\n    insertionRangeCheck(index)\n\n    if (index == size) return addAll(elements)\n    if\n    (elements.isEmpty()) return false\n    when (index) {\n      size -> return addAll(elements)\n      0 -> array\n    = elements.toArray<Any?>() + array\n      else -> array = array.copyOfRange(0,\n    index).asDynamic().concat(elements.toArray<Any?>(), array.copyOfRange(index, size))\n    }\n\n    modCount++\n    return true\n  }\n\n  actual override fun removeAt(index: Int): E {\n    checkIsMutable()\n    rangeCheck(index)\n    modCount++\n    return if (index == lastIndex)\n      array.asDynamic().pop()\n    else\n      array.asDynamic().splice(index, 1)[0]\n  }\n\n  actual override fun remove(element: E): Boolean {\n\n    checkIsMutable()\n    for (index in array.indices) {\n      if (array[index] == element) {\n\n    array.asDynamic().splice(index, 1)\n      modCount++\n\n      return true\n    }\n  }\n  return false\n}\n\n  override fun removeRange(fromIndex: Int,\n  toIndex: Int) {\n    checkIsMutable()\n    modCount++\n    array.asDynamic().splice(fromIndex, toIndex -\n  fromIndex)\n  }\n\n  actual override fun clear() {\n    checkIsMutable()\n    array = emptyArray()\n    modCount++\n  }\n\n  actual override fun indexOf(element: E): Int = array.indexOf(element)\n\n  actual\n  override fun lastIndexOf(element: E): Int = array.lastIndexOf(element)\n\n  override fun toString() =\n  arrayToString(array)\n\n  @Suppress("UNCHECKED_CAST")\n  override fun <T> toArray(array: Array<T>):\n  Array<T> {\n    if (array.size < size) {\n      return toArray() as Array<T>\n    }\n    (this.array as\n  Array<T>).copyInto(array)\n    if (array.size > size) {\n      array[size] = null as T // null-terminate\n    }\n    return array\n  }\n\n  override fun\n  toArray(): Array<Any?> {\n    return js("[]").slice.call(array)\n  }\n\n  internal override fun\n  checkIsMutable() {\n    if (isReadOnly) throw UnsupportedOperationException()\n  }\n\n  private fun\n  rangeCheck(index: Int) = index.apply {\n    AbstractList.checkElementIndex(index, size)\n  }\n\n  private fun\n  insertionRangeCheck(index: Int) = index.apply {\n    AbstractList.checkPositionIndex(index, size)\n  }\n  }", "/*\n  *\n  * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n  *\n  * Use of this source code\n  is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n  */\n  }\n\n  package\n  kotlin.collections\n\n  internal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {\n    if\n    (getStableSortingIsSupported()) {\n      array.asDynamic().sort(comparison)\n    } else {\n      mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))\n    }\n  }\n\n  internal fun\n  <T> sortArrayWith(array:\n  Array<out T>, comparator: Comparator<in T>) {\n    if (getStableSortingIsSupported()) {\n      val comparison = {\n    a: T, b: T -> comparator.compare(a, b)\n    }\n    array.asDynamic().sort(comparison)\n  } else {\n  }

```

```

mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)\n }\n}\n\ninternal fun <T>
sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {\n if
(fromIndex < toIndex - 1) {\n mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex - 1, comparator)\n
}\n}\n\ninternal fun <T : Comparable<T>> sortArray(array: Array<out T>) {\n if
(getStableSortingIsSupported()) {\n val comparison = { a: T, b: T -> a.compareTo(b) }\n
array.asDynamic().sort(comparison)\n } else {\n mergeSort(array.unsafeCast<Array<T>>(), 0,
array.lastIndex, naturalOrder())\n }\n}\n\nprivate var _stableSortingIsSupported: Boolean? = null\nprivate fun
getStableSortingIsSupported(): Boolean
{\n _stableSortingIsSupported?.let { return it }\n _stableSortingIsSupported = false\n\n val array =
js("[ ]").unsafeCast<Array<Int>>()\n // known implementations may use stable sort for arrays of up to 512
elements\n // so we create slightly more elements to test stability\n for (index in 0 until 600)
array.asDynamic().push(index)\n val comparison = { a: Int, b: Int -> (a and 3) - (b and 3) }\n
array.asDynamic().sort(comparison)\n for (index in 1 until array.size) {\n val a = array[index - 1]\n val b
= array[index]\n if ((a and 3) == (b and 3) && a >= b) return false\n }\n _stableSortingIsSupported = true\n
return true\n}\n\nprivate fun <T> mergeSort(array: Array<T>, start: Int, endInclusive: Int, comparator:
Comparator<in T>) {\n val buffer = arrayOfNulls<Any?>(array.size).unsafeCast<Array<T>>()\n val result =
mergeSort(array, buffer, start, endInclusive, comparator)\n if (result !== array) {\n for (i
in start..endInclusive) array[i] = result[i]\n }\n}\n\n// Both start and end are inclusive indices.\nprivate fun <T>
mergeSort(array: Array<T>, buffer: Array<T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {\n
if (start == end) {\n return array\n }\n\n val median = (start + end) / 2\n val left = mergeSort(array, buffer,
start, median, comparator)\n val right = mergeSort(array, buffer, median + 1, end, comparator)\n val target = if
(left === buffer) array else buffer\n\n // Merge.\n var leftIndex = start\n var rightIndex = median + 1\n for (i
in start..end) {\n when {\n leftIndex <= median && rightIndex <= end -> {\n val leftValue =
left[leftIndex]\n val rightValue = right[rightIndex]\n if (comparator.compare(leftValue,
rightValue) <= 0) {\n target[i] = leftValue\n leftIndex++\n } else {\n
target[i]
= rightValue\n rightIndex++\n }\n }\n leftIndex <= median -> {\n
target[i] = left[leftIndex]\n leftIndex++\n }\n else /* rightIndex <= end */ -> {\n
target[i] = right[rightIndex]\n rightIndex++\n Unit // TODO: Fix KT-31506\n }\n }\n
}\n\n return target\n}, "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.collections\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@kotlin.js.JsName("\ncontentDeepHashCodeImpl")\ninternal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {\n if (this ==
null) return 0\n var result = 1\n for (element in this) {\n val elementHash = when {\n element
== null -> 0\n isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()\n\n
element is UByteArray -> element.contentHashCode()\n element is UShortArray ->
element.contentHashCode()\n element is UIntArray -> element.contentHashCode()\n element is
ULongArray -> element.contentHashCode()\n\n else -> element.hashCode()\n }\n\n result = 31 * result + elementHash\n }\n return result\n}, "/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal interface
EqualityComparator {\n /**\n * Subclasses must override to return a value indicating\n * whether or not two
keys or values are equal.\n */\n abstract fun equals(value1: Any?, value2: Any?): Boolean\n\n /**\n * Subclasses must override to return the hash code of a given key.\n */\n abstract fun
getHashCode(value: Any?): Int\n\n object HashCode : EqualityComparator {\n override fun equals(value1:
Any?, value2: Any?): Boolean = value1 == value2\n\n override fun getHashCode(value: Any?): Int =
value?.hashCode() ?: 0\n }\n}, "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language

```

contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n * Based on GWT AbstractHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap] interface.\n * This implementation makes no guarantees regarding the order of enumeration of [keys], [values] and [entries] collections.\n */\n// Classes that extend HashMap and implement `build()`

(freezing) operation\n// have to make sure mutating methods check `checkIsMutable`.\n\npublic actual open class HashMap<K, V> : AbstractMutableMap<K, V>, MutableMap<K, V> {\n\n private inner class EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n\n override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")\n\n override fun clear() {\n\n this@HashMap.clear()\n\n }\n\n override fun containsEntry(element: Map.Entry<K, V>): Boolean = this@HashMap.containsEntry(element)\n\n override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = internalMap.iterator()\n\n\n override fun removeEntry(element: Map.Entry<K, V>): Boolean {\n\n if (contains(element)) {\n\n this@HashMap.remove(element.key)\n\n return true\n\n }\n\n return false\n\n }\n\n\n override val size: Int get() =

this@HashMap.size\n

\n\n\n /**\n * Internal implementation of the map: either string-based or hashCode-based.\n */\n\n private val internalMap: InternalMap<K, V>\n\n private val equality: EqualityComparator\n\n\n internal constructor(internalMap: InternalMap<K, V>) : super() {\n\n this.internalMap = internalMap\n\n this.equality = internalMap.equality\n\n }\n\n\n /**\n * Constructs an empty [HashMap] instance.\n */\n\n actual constructor() : this(InternalHashMap(EqualityComparator.HashCode))\n\n\n /**\n * Constructs an empty [HashMap] instance.\n */\n\n * @param initialCapacity the initial capacity (ignored)\n * @param loadFactor the load factor (ignored)\n * @throws IllegalArgumentException if the initial capacity or load factor are negative\n */\n\n actual constructor(initialCapacity: Int, loadFactor: Float) : this() {\n\n // This implementation of HashMap has no need of load factors or capacities.\n\n require(initialCapacity

>= 0) { "Negative initial capacity: \$initialCapacity" }\n\n require(loadFactor >= 0) { "Non-positive load factor: \$loadFactor" }\n\n }\n\n\n actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n\n /**\n * Constructs an instance of [HashMap] filled with the contents of the specified [original] map.\n */\n\n actual

constructor(original: Map<out K, V>) : this() {\n\n this.putAll(original)\n\n }\n\n\n actual override fun clear() {\n\n internalMap.clear()\n\n // structureChanged(this)\n\n }\n\n\n actual override fun containsKey(key: K): Boolean = internalMap.containsKey()\n\n\n actual override fun containsValue(value: V): Boolean = internalMap.any { equality.equals(it.value, value) }\n\n\n private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? = null\n\n\n actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n\n\n get() {\n\n if (_entries == null) {\n\n _entries

= createEntrySet()\n\n }\n\n\n return _entries!!\n\n }\n\n\n internal open fun createEntrySet():

MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n\n actual override operator fun get(key: K): V? = internalMap.get(key)\n\n\n actual override fun put(key: K, value: V): V? = internalMap.put(key, value)\n\n\n\n actual override fun remove(key: K): V? = internalMap.remove(key)\n\n\n\n actual override val size: Int get() =

internalMap.size\n\n}\n\n\n/**\n * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing them.\n */\n\npublic fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {\n\n return HashMap<String,

V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }\n\n}\n\n",\n\n*/\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found

in the license/LICENSE.txt file.\n */\n\n * Based on GWT HashSet\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * The implementation of the [MutableSet] interface, backed by a [HashMap] instance.\n */\n// Classes that extend HashSet and implement `build()` (freezing) operation\n// have to make sure mutating methods check `checkIsMutable`.\n\npublic actual open class HashSet<E> :

```

AbstractMutableSet<E>, MutableSet<E> {\n\n    internal val map: HashMap<E, Any>\n\n    /**\n     * Constructs a new empty [HashSet].\n     */\n    actual constructor() {\n        map = HashMap<E, Any>()\n    }\n\n    /**\n     * Constructs a new [HashSet] filled with the elements of the specified collection.\n     */\n    actual constructor(elements: Collection<E>) {\n        map = HashMap<E, Any>(elements.size)\n        addAll(elements)\n    }\n\n    /**\n     * Constructs a new empty [HashSet].\n     */\n    * @param initialCapacity the initial capacity (ignored)\n    * @param loadFactor the load factor (ignored)\n    * @throws IllegalArgumentException if the initial capacity or load factor are negative\n    */\n    actual constructor(initialCapacity: Int, loadFactor: Float) {\n        map = HashMap<E, Any>(initialCapacity, loadFactor)\n    }\n\n    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n    /**\n     * Protected constructor to specify the underlying map. This is used by\n     * LinkedHashMap.\n     */\n    * @param map underlying map to use.\n    */\n    internal constructor(map: HashMap<E, Any>) {\n        this.map = map\n    }\n\n    actual override fun add(element: E): Boolean {\n        val old = map.put(element, this)\n        return old == null\n    }\n\n    actual override fun clear() {\n        map.clear()\n    }\n\n    public override fun clone(): Any {\n        return HashSet<E>(this)\n    }\n\n    actual override operator fun contains(element: E): Boolean = map.containsKey(element)\n\n    actual override fun isEmpty(): Boolean = map.isEmpty()\n\n    actual override fun iterator(): MutableIterator<E> = map.keys.iterator()\n\n    actual override fun remove(element: E): Boolean = map.remove(element) != null\n\n    actual override val size: Int get() = map.size\n}\n\n/**\n * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,\n * which elements the keys as properties of JS object without hashing them.\n */\npublic fun stringSetOf(vararg elements: String): HashSet<String> {\n    return HashSet(stringMapOf<Any>()).apply { addAll(elements) }\n}\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n * Based on GWT InternalHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\nimport kotlin.collections.AbstractMutableMap.SimpleEntry\n\n/**\n * A simple wrapper around JavaScriptObject to provide [java.util.Map]-like semantics for any\n * key type.\n */\n\n * Implementation notes:\n * A key's hashCode is the index in backingMap which should contain that key. Since several keys may\n * have the same hash, each value in hashCodeMap is actually an array containing all entries whose\n * keys share the same hash.\n */\n\ninternal class InternalHashMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n    private var backingMap: dynamic = createJsMap()\n    override var size: Int = 0\n    private set\n\n    override fun put(key: K, value: V): V? {\n        val hashCode = equality.getHashCode(key)\n        val chainOrEntry = getChainOrEntryOrNull(hashCode)\n        if (chainOrEntry == null) {\n            // This is a new chain, put it to the map.\n            backingMap[hashCode] = SimpleEntry(key, value)\n        } else {\n            if (chainOrEntry !is Array<*>) {\n                // It is an entry\n                val entry: SimpleEntry<K, V> = chainOrEntry\n                if (equality.equals(entry.key, key)) {\n                    return entry.setValue(value)\n                }\n            } else {\n                backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))\n                size++\n            }\n            return null\n        }\n    }\n\n    } else {\n        // Chain already exists, perhaps key also exists.\n        val chain: Array<MutableEntry<K, V>> = chainOrEntry\n        val entry = chain.findEntryInChain(key)\n        if (entry != null) {\n            return entry.setValue(value)\n        }\n        chain.asDynamic().push(SimpleEntry(key, value))\n        size++\n        structureChanged(host)\n        return null\n    }\n\n    override fun remove(key: K): V? {\n        val hashCode = equality.getHashCode(key)\n        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null\n        if (chainOrEntry !is Array<*>) {\n            val entry: MutableEntry<K, V> = chainOrEntry\n            if (equality.equals(entry.key, key)) {\n                jsDeleteProperty(backingMap, hashCode)\n                size--\n                return entry.value\n            } else {\n                return null\n            }\n        } else {\n            val chain: Array<MutableEntry<K, V>> = chainOrEntry\n            for (index in chain.indices) {\n                val entry = chain[index]\n                if (equality.equals(key, entry.key)) {\n                    if (chain.size == 1) {\n                        chain.asDynamic().length = 0\n                        // remove the whole

```

```

array\n                jsDeleteProperty(backingMap, hashCode)\n                } else {\n                // splice out
the entry we're removing\n                chain.asDynamic().splice(index,
1)\n                }\n                size--\n//                structureChanged(host)\n                return entry.value\n
}\n                }\n                return null\n }\n\n override fun clear() {\n    backingMap = createJsMap()\nsize = 0\n }\n\n override fun contains(key: K): Boolean = getEntry(key) != null\n\n override fun get(key: K):
V? = getEntry(key)?.value\n\n private fun getEntry(key: K): MutableEntry<K, V>? {\n    val chainOrEntry =
getChainOrEntryOrNull(equality.getHashCode(key)) ?: return null\n    if (chainOrEntry != Array<*>) {\n
val entry: MutableEntry<K, V> = chainOrEntry\n        if (equality.equals(entry.key, key)) {\n            return
entry\n        } else {\n            return null\n        }\n    } else {\n        val chain: Array<MutableEntry<K,
V>> = chainOrEntry\n            return chain.findEntryInChain(key)\n        }\n    }\n\n private fun
Array<MutableEntry<K, V>>.findEntryInChain(key: K): MutableEntry<K, V>? =\n    firstOrNull { entry ->
equality.equals(entry.key, key) }\n\n override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n\n
return object : MutableIterator<MutableEntry<K, V>> {\n        var state = -1 // -1 not ready, 0 - ready, 1 -
done\n\n        val keys: Array<String> = js("Object").keys(backingMap)\n        var keyIndex = -1\n\n
var chainOrEntry: dynamic = null\n        var isChain = false\n        var itemIndex = -1\n        var lastEntry:
MutableEntry<K, V>? = null\n\n        private fun computeNext(): Int {\n            if (chainOrEntry != null &&
isChain) {\n                val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size\n
                if (++itemIndex < chainSize)\n                    return 0\n            }\n            if (++keyIndex < keys.size)
{\n                chainOrEntry
= backingMap[keys[keyIndex]]\n                isChain = chainOrEntry is Array<*>\n                itemIndex = 0\n
                return 0\n            } else {\n                chainOrEntry = null\n                return 1\n            }\n
}\n\n        override fun hasNext(): Boolean {\n            if (state == -1)\n                state = computeNext()\n
                return state == 0\n            }\n\n        override fun next(): MutableEntry<K, V> {\n            if (!hasNext())
throw NoSuchElementException()\n                val lastEntry = if (isChain) {\n                    chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]\n                } else {\n                    chainOrEntry.unsafeCast<MutableEntry<K, V>>()\n                }\n                this.lastEntry = lastEntry\n
                state = -1\n                return lastEntry\n            }\n\n            override fun remove() {\n
                checkNotNull(lastEntry)\n                this@InternalHashMap.remove(lastEntry!!.key)\n                lastEntry =
null\n                // the chain being iterated just got modified by InternalHashMap.remove\n                itemIndex-
-1\n            }\n        }\n    }\n\n private fun getChainOrEntryOrNull(hashCode: Int): dynamic {\n    val
chainOrEntry = backingMap[hashCode]\n    return if (chainOrEntry === undefined) null else chainOrEntry\n
}\n\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.collections\n/**\n * The common interface of [InternalStringMap] and
[InternalHashMap].\n */\n\ninternal interface InternalMap<K, V> :
MutableIterable<MutableMap.MutableEntry<K, V>> {\n    val equality: EqualityComparator\n    val size: Int\n
operator fun contains(key: K): Boolean\n
operator fun get(key: K): V?\n\n    fun put(key: K, value: V): V?\n    fun remove(key: K): V?\n    fun clear():
Unit\n\n    fun createJsMap(): dynamic {\n        val result = js("Object.create(null)")\n        // force to switch object
representation to dictionary mode\n        result["foo"] = 1\n        jsDeleteProperty(result, "foo")\n        return
result\n    }\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n/*\n * Based on GWT InternalStringMap\n * Copyright 2008 Google Inc.\n */\n\npackage
kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * A simple wrapper around
JavaScript Map for key type is string.\n */\n * Though this map is instantiated only with K=String, the K type is not
fixed to String statically,\n * because we want to have it erased to Any? in order not to
generate type-safe override bridges for\n * [get], [contains], [remove] etc, if they ever are generated.\n */\n\ninternal
class InternalStringMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n\n    private var

```

```

backingMap: dynamic = createJsMap()\n  override var size: Int = 0\n    private set\n\n// /**\n//  * A mod
count to track 'value' replacements in map to ensure that the 'value' that we have in the\n//  * iterator entry is
guaranteed to be still correct.\n//  * This is to optimize for the common scenario where the values are not modified
during\n//  * iterations where the entries are never stale.\n//  */\n\n// private var valueMod: Int = 0\n  override
operator fun contains(key: K): Boolean {\n    if (key !is String) return false\n    return backingMap[key] !==
undefined\n  }\n\n  override operator fun get(key: K): V? {\n    if (key !is String) return null\n    val value =
backingMap[key]\n    return if (value
!== undefined) value.unsafeCast<V>() else null\n  }\n\n  override fun put(key: K, value: V): V? {\n
require(key is String)\n    val oldValue = backingMap[key]\n    backingMap[key] = value\n\n    if (oldValue
=== undefined) {\n        size++\n//        structureChanged(host)\n        return null\n    } else {\n//
valueMod++\n        return oldValue.unsafeCast<V>()\n    }\n  }\n\n  override fun remove(key: K): V? {\n
if (key !is String) return null\n    val value = backingMap[key]\n    if (value !== undefined) {\n
jsDeleteProperty(backingMap, key)\n        size--\n//        structureChanged(host)\n        return
value.unsafeCast<V>()\n    } else {\n//        valueMod++\n        return null\n    }\n  }\n\n  override fun
clear() {\n    backingMap = createJsMap()\n    size = 0\n  }\n\n  override fun iterator():
MutableIterator<MutableEntry<K, V>> {\n
    return object : MutableIterator<MutableEntry<K, V>> {\n        private val keys: Array<String> =
js("Object").keys(backingMap)\n        private val iterator = keys.iterator()\n        private var lastKey: String? =
null\n\n        override fun hasNext(): Boolean = iterator.hasNext()\n\n        override fun next():
MutableEntry<K, V> {\n            val key = iterator.next()\n            lastKey = key\n
@Suppress("UNCHECKED_CAST")\n            return newMapEntry(key as K)\n        }\n\n        override
fun remove() {\n            @Suppress("UNCHECKED_CAST")\n            this@InternalStringMap.remove(checkNotNull(lastKey) as K)\n        }\n    }\n\n    private fun
newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {\n        override val key: K get() =
key\n        override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()\n        override fun
setValue(newValue:
V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()\n        override fun hashCode(): Int =
AbstractMap.entryHashCode(this)\n        override fun toString(): String = AbstractMap.entryToString(this)\n
        override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this, other)\n    }\n}\n\n", "/*\n * Copyright
2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT
LinkedHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport
kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap]
interface, which additionally preserves the insertion order\n * of entries during the iteration.\n * The insertion
order is preserved by maintaining a doubly-linked list of all of its entries.\n */\n\npublic actual open class
LinkedHashMap<K, V> : HashMap<K,
V>, MutableMap<K, V> {\n\n    /**\n     * The entry we use includes next/prev pointers for a doubly-linked
circular\n     * list with a head node. This reduces the special cases we have to deal with\n     * in the list
operations.\n     * Note that we duplicate the key from the underlying hash map so we can find\n     * the eldest
entry. The alternative would have been to modify HashMap so more\n     * of the code was directly usable here, but
this would have added some\n     * overhead to HashMap, or to reimplement most of the HashMap code here with\n
     * small modifications. Paying a small storage cost only if you use\n     * LinkedHashMap and minimizing code size
seemed like a better tradeoff\n     */\n    private inner class ChainEntry<K, V>(key: K, value: V) :
AbstractMutableMap.SimpleEntry<K, V>(key, value) {\n        internal var next: ChainEntry<K, V>? = null\n
        internal var prev: ChainEntry<K, V>? = null\n\n        override fun setValue(newValue: V): V {\n
            this@LinkedHashMap.checkIsMutable()\n            return super.setValue(newValue)\n        }\n    }\n\n    private inner class EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n        private inner class
EntryIterator : MutableIterator<MutableEntry<K, V>> {\n            // The last entry that was returned from this

```



```

iterator.\n        private var last: ChainEntry<K, V>? = null\n\n        // The next entry to return from this
iterator.\n        private var next: ChainEntry<K, V>? = null\n\n        init {\n            next = head\n//
recordLastKnownStructure(map, this)\n        }\n\n        override fun hasNext(): Boolean {\n            return
next != null\n        }\n\n        override fun next(): MutableEntry<K, V> {\n//
checkStructuralChange(map, this)\n            if (!hasNext()) throw NoSuchElementException()\n\n            val
current = next!!\n            last = current\n\n            next = current.next.takeIf { it != head }\n            return current\n        }\n\n        override fun remove()
{\n            check(last != null)\n            this@EntrySet.checkIsMutable()\n//
checkStructuralChange(map, this)\n            last!!.remove()\n            map.remove(last!!.key)\n//
recordLastKnownStructure(map, this)\n            last = null\n        }\n\n        override fun add(element:
MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")\n
override fun clear() {\n            this@LinkedHashMap.clear()\n        }\n\n        override fun containsEntry(element:
Map.Entry<K, V>): Boolean = this@LinkedHashMap.containsEntry(element)\n\n        override operator fun
iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()\n\n        override fun removeEntry(element:
Map.Entry<K, V>): Boolean {\n            checkIsMutable()\n
            if (contains(element)) {\n                this@LinkedHashMap.remove(element.key)\n                return true\n
            }\n            return false\n        }\n\n        override val size: Int get() = this@LinkedHashMap.size\n\n        override
fun checkIsMutable(): Unit = this@LinkedHashMap.checkIsMutable()\n        }\n\n        /*\n         * The head of the insert
order chain, which is a doubly-linked circular\n         * list.\n         * The most recently inserted node is at the end of the
chain, ie.\n         * chain.prev.\n         */\n        private var head: ChainEntry<K, V>? = null\n\n        /*\n         * Add this node to
the end of the chain.\n         */\n        private fun ChainEntry<K, V>.addToEnd() {\n            // This entry is not in the list.\n
check(next == null && prev == null)\n            val _head = head\n            if (_head == null) {\n                head = this\n
                next = this\n                prev = this\n            } else {\n                // Chain is valid.\n                val _tail =
checkNotNull(_head.prev)\n
                // Update me.\n                prev = _tail\n                next = _head\n                // Update my new siblings: current head
and old tail\n                _head.prev = this\n                _tail.next = this\n            }\n        }\n\n        /*\n         * Remove this node from
the chain it is a part of.\n         */\n        private fun ChainEntry<K, V>.remove() {\n            if (this.next === this) {\n
// if this is single element, remove head\n                head = null\n            } else {\n                if (head === this) {\n
// if this is first element, move head to next\n                    head = next\n                }\n                next!!.prev = prev\n
                prev!!.next = next\n            }\n            next = null\n            prev = null\n        }\n\n        /*\n         * The hashmap that keeps track of
our entries and the chain. Note that we\n         * duplicate the key here to eliminate changes to HashMap and minimize
the\n         * code here, at the expense of additional space.\n         */\n        private
val map: HashMap<K, ChainEntry<K, V>>\n        private var isReadOnly: Boolean = false\n\n        /*\n         *
Constructs an empty [LinkedHashMap] instance.\n         */\n        actual constructor() : super() {\n            map =
HashMap<K, ChainEntry<K, V>>()\n        }\n\n        internal constructor(backingMap: HashMap<K, Any>) : super() {\n
            @Suppress("UNCHECKED_CAST") // expected to work due to erasure\n            map = backingMap as
HashMap<K, ChainEntry<K, V>>\n        }\n\n        /*\n         * Constructs an empty [LinkedHashMap] instance.\n         */\n
        * @param initialCapacity the initial capacity (ignored)\n         * @param loadFactor the load factor (ignored)\n
        * @throws IllegalArgumentException if the initial capacity or load factor are negative\n         */\n        actual
constructor(initialCapacity: Int, loadFactor: Float) : super(initialCapacity, loadFactor) {\n            map = HashMap<K,
ChainEntry<K, V>>()\n        }\n\n        actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n
        /*\n         * Constructs an instance of [LinkedHashMap] filled with the contents of the specified [original] map.\n
        */\n        * @param original the original map\n         */\n        actual constructor(original: Map<out K, V>) {\n            map =
HashMap<K, ChainEntry<K, V>>()\n            this.putAll(original)\n        }\n\n        @PublishedApi\n        internal fun build(): Map<K, V> {\n            checkIsMutable()\n
            isReadOnly = true\n            return this\n        }\n\n        actual override fun clear() {\n            checkIsMutable()\n
            map.clear()\n            head = null\n        }\n\n        // override fun clone(): Any {\n//            return LinkedHashMap(this)\n//
        }\n\n        actual override fun containsKey(key: K): Boolean = map.containsKey(key)\n\n        actual override fun
containsValue(value: V): Boolean {\n            var node: ChainEntry<K, V> = head ?: return false\n            do {\n                if

```

```

(node.value == value) {\n        return true\n    }\n    node = node.next!!\n    } while (node !==
head)\n    return false\n }\n\n
internal override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n actual
override operator fun get(key: K): V? = map.get(key)?.value\n\n actual override fun put(key: K, value: V): V? {\n
    checkIsMutable()\n\n    val old = map.get(key)\n    if (old == null) {\n        val newEntry =
ChainEntry(key, value)\n        map.put(key, newEntry)\n        newEntry.addToEnd()\n        return null\n
    } else {\n        return old.setValue(value)\n    }\n }\n\n actual override fun remove(key: K): V? {\n
checkIsMutable()\n\n    val entry = map.remove(key)\n    if (entry != null) {\n        entry.remove()\n
return entry.value\n    }\n    return null\n }\n\n actual override val size: Int get() = map.size\n\n internal
override fun checkIsMutable() {\n    if (isReadOnly) throw UnsupportedOperationException()\n }\n}\n\n/**\n *
Constructs the specialized implementation
of [LinkedHashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing
them.\n */\n\n public fun <V> linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {\n
return LinkedHashMap<String, V>(stringMapOf<Any>()).apply { putAll(pairs) }\n }\n\n /**\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n /**\n * Based on GWT LinkedHashMap\n
* Copyright 2008 Google Inc.\n */\n\n package kotlin.collections\n\n /**\n * The implementation of the [MutableSet]
interface, backed by a [LinkedHashMap] instance.\n * This implementation preserves the insertion order of
elements during the iteration.\n */\n\n public actual open class LinkedHashMap<E> : HashSet<E>, MutableSet<E>
{\n\n    internal constructor(map: LinkedHashMap<E, Any>) : super(map)\n\n    /**\n     * Constructs a new empty
[LinkedHashSet].\n     */\n\n    actual constructor() : super(LinkedHashMap<E, Any>())\n\n    /**\n     * Constructs a
new [LinkedHashSet] filled with the elements of the specified collection.\n     */\n\n    actual constructor(elements:
Collection<E>) : super(LinkedHashMap<E, Any>()) {\n        addAll(elements)\n    }\n\n    /**\n     * Constructs a
new empty [LinkedHashSet].\n     */\n\n    /**\n     * @param initialCapacity the initial capacity (ignored)\n     * @param
loadFactor the load factor (ignored)\n     * @throws IllegalArgumentException if the initial capacity or
load factor are negative\n     */\n\n    actual constructor(initialCapacity: Int, loadFactor: Float) :
super(LinkedHashMap<E, Any>(initialCapacity, loadFactor))\n\n    actual constructor(initialCapacity: Int) :
this(initialCapacity, 0.0f)\n\n    @PublishedApi\n    internal fun build(): Set<E> {\n        (map as
LinkedHashMap<E, Any>).build()\n        return this\n    }\n\n    internal override fun checkIsMutable():
Unit = map.checkIsMutable()\n\n // public override fun clone(): Any {\n //     return LinkedHashMap(this)\n //
}\n\n }\n\n /**\n * Creates a new instance of the specialized implementation of [LinkedHashSet] with the specified
[String] elements,\n * which elements the keys as properties of JS object without hashing them.\n */\n\n public fun
linkedStringSetOf(vararg elements: String): LinkedHashSet<String> {\n    return
LinkedHashSet(linkedStringMapOf<Any>()).apply { addAll(elements) }\n }\n\n /**\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n package kotlin\n\n import
kotlin.contracts.*\n\n @DeprecatedSinceKotlin(warningSince = \"1.6\")\n @Deprecated(\"Synchronization on any
object is not supported in Kotlin/JS\",
ReplaceWith(\"run(block)\"))\n @kotlin.internal.InlineOnly\n @Suppress(\"UNUSED_PARAMETER\")\n public
inline fun <R>
synchronized(lock: Any, block: () -> R): R {\n    contract {\n        callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n    }\n    return block()\n }\n\n /**\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n package kotlin.io\n\n internal abstract class BaseOutput {\n
open fun println() {\n    print(\"\\n\")\n }\n\n open fun println(message: Any?) {\n    print(message)\n
println()\n }\n\n abstract fun print(message: Any?)\n\n open fun flush() {\n }\n\n /**\n * JsName used to make the
declaration available outside of module to test it\n */\n @JsName(\"NodeJsOutput\")\n internal class NodeJsOutput(val
outputStream: dynamic) : BaseOutput() {\n    override fun print(message: Any?) {\n        // TODO: Using local

```

```

variable because of bug in block decomposition lowering in IR backend\n    val messageString =
String(message)\n    outputStream.write(messageString)\n    }\n}\n\n/** JsName used to make the declaration
available outside of module to test it */\n@JsName("OutputToConsoleLog")\ninternal class OutputToConsoleLog
: BaseOutput() {\n    override fun print(message: Any?) {\n        console.log(message)\n    }\n\n    override fun
println(message: Any?) {\n        console.log(message)\n    }\n\n    override fun println() {\n        console.log("\\")\n
}\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in
*/\n@JsName("BufferedOutput")\ninternal open class BufferedOutput : BaseOutput() {\n    var buffer = ""\n\n    override fun print(message: Any?) {\n        buffer += String(message)\n    }\n\n    override fun flush() {\n        buffer
= ""\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it
*/\n@JsName("BufferedOutputToConsoleLog")\ninternal class BufferedOutputToConsoleLog :
BufferedOutput() {\n    override fun print(message: Any?) {\n        var s = String(message)\n        val i =
s.nativeLastIndexOf("\\n", 0)\n        if (i >= 0) {\n            buffer += s.substring(0, i)\n            flush()\n            s =
s.substring(i + 1)\n        }\n        buffer += s\n    }\n\n    override fun flush() {\n        console.log(buffer)\n        buffer
= ""\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at
try.kotl.in */\n@JsName("output")\ninternal var output = run {\n    val isNode: Boolean = js("typeof process !==
'undefined' && process.versions && !!process.versions.node")\n    if (isNode) NodeJsOutput(js("process.stdout"))
else BufferedOutputToConsoleLog()\n}\n\n@kotlin.internal.InlineOnly\nprivate inline fun String(value: Any?):
String = js("String")(value)\n\n/** Prints the line separator to the standard output stream. */\npublic actual fun
println() {\n    output.println()\n}\n\n/** Prints the
given [message] and the line separator to the standard output stream. */\npublic actual fun println(message: Any?)
{\n    output.println(message)\n}\n\n/** Prints the given [message] to the standard output stream. */\npublic actual
fun print(message: Any?) {\n    output.print(message)\n}\n\n@SinceKotlin("1.6")\npublic actual fun readln():
String = throw UnsupportedOperationException("readln is not supported in
Kotlin/JS")\n\n@SinceKotlin("1.6")\npublic actual fun readlnOrNull(): String? = throw
UnsupportedOperationException("readlnOrNull is not supported in Kotlin/JS"), /*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport
kotlin.coroutines.intrinsics.CoroutineSingletons.*\nimport
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal
actual
class SafeContinuation<in T>\ninternal actual constructor(\n    private val delegate: Continuation<T>,\n    initialResult: Any?\n) : Continuation<T> {\n    @PublishedApi\n    internal actual constructor(delegate:
Continuation<T>) : this(delegate, UNDECIDED)\n\n    public actual override val context: CoroutineContext\n    get() = delegate.context\n\n    private var result: Any? = initialResult\n\n    public actual override fun
resumeWith(result: Result<T>) {\n        val cur = this.result\n        when {\n            cur === UNDECIDED -> {\n
                this.result = result.value\n            }\n            cur === COROUTINE_SUSPENDED -> {\n                this.result =
RESUMED\n                delegate.resumeWith(result)\n            }\n            else -> throw
IllegalStateException("Already resumed")\n        }\n    }\n\n    @PublishedApi\n    internal actual fun
getOrThrow(): Any? {\n        if (result === UNDECIDED) {\n            result = COROUTINE_SUSPENDED\n            return
COROUTINE_SUSPENDED\n        }\n        val result = this.result\n        return when {\n            result ===
RESUMED -> COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED
upstream\n            result is Result.Failure -> throw result.exception\n            else -> result // either
COROUTINE_SUSPENDED or data\n        }\n    }\n}\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.coroutines.cancellation\n\n@SinceKotlin("1.4")\npublic actual open class CancellationException :
IllegalStateException {\n    actual constructor(): super()\n    actual constructor(message: String?): super(message)\n
    constructor(message: String?, cause: Throwable?): super(message, cause)\n    constructor(cause: Throwable?):

```

```

super(cause)\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines.js.internal\n\nimport
kotlin.coroutines.Continuation\n\nimport
kotlin.coroutines.EmptyCoroutineContext\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal val
EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result ->\n result.getOrThrow()\n}"/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n/**\n * Exposes the [Date API](https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.\n
*\n\n@Suppress("NOT_DOCUMENTED")\npublic external class Date() {\n public constructor(milliseconds:
Number)\n\n public constructor(dateString: String)\n\n public constructor(year:
Int, month: Int)\n\n public constructor(year: Int, month: Int, day: Int)\n\n public constructor(year: Int, month:
Int, day: Int, hour: Int)\n\n public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int)\n\n public
constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int)\n\n public constructor(year: Int,
month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number)\n\n public fun getDate(): Int\n\n
public fun getDay(): Int\n\n public fun getFullYear(): Int\n\n public fun getHours(): Int\n\n public fun
getMilliseconds(): Int\n\n public fun getMinutes(): Int\n\n public fun getMonth(): Int\n\n public fun
getSeconds(): Int\n\n public fun getTime(): Double\n\n public fun getTimezoneOffset(): Int\n\n public fun
getUTCDate(): Int\n\n public fun getUTCDay(): Int\n\n public fun getUTCFullYear(): Int\n\n public fun
getUTCHours(): Int\n\n public fun getUTCMilliseconds():
Int\n\n public fun getUTCMinutes(): Int\n\n public fun getUTCMonth(): Int\n\n public fun getUTCSeconds():
Int\n\n public fun toDateString(): String\n\n public fun toISOString(): String\n\n public fun toJSON(): Json\n\n
public fun toLocaleDateString(locales: Array<String> = definedExternally, options: LocaleOptions =
definedExternally): String\n\n public fun toLocaleDateString(locales: String, options: LocaleOptions =
definedExternally): String\n\n public fun toLocaleString(locales: Array<String> = definedExternally, options:
LocaleOptions = definedExternally): String\n\n public fun toLocaleString(locales: String, options: LocaleOptions
= definedExternally): String\n\n public fun toLocaleTimeString(locales: Array<String> = definedExternally,
options: LocaleOptions = definedExternally): String\n\n public fun toLocaleTimeString(locales: String, options:
LocaleOptions = definedExternally): String\n\n public fun toTimeString(): String\n\n public
fun toUTCString(): String\n\n public companion object {\n public fun now(): Double\n\n public fun
parse(dateString: String): Double\n\n public fun UTC(year: Int, month: Int): Double\n\n public fun
UTC(year: Int, month: Int, day: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int):
Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int): Double\n\n public fun
UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int): Double\n\n public fun UTC(year: Int,
month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number): Double\n } \n\n public interface
LocaleOptions {\n public var localeMatcher: String?\n\n public var timeZone: String?\n\n public var
hour12: Boolean?\n\n public var formatMatcher: String?\n\n public var weekday: String?\n\n public var
era: String?\n\n public var year: String?\n\n
public var month: String?\n\n public var day: String?\n\n public var hour: String?\n\n public var
minute: String?\n\n public var second: String?\n\n public var timeZoneName: String?\n } \n\n\npublic
inline fun dateLocaleOptions(init: Date.LocaleOptions.() -> Unit): Date.LocaleOptions {\n val result = js("new
Object()\").unsafeCast<Date.LocaleOptions>()\n init(result)\n return result\n}"/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport
org.w3c.dom.Document\n\nimport org.w3c.dom.Element\n\nimport
kotlin.internal.LowPriorityInOverloadResolution\n\nimport kotlinx.dom.appendElement as
newAppendElement\n\nimport kotlinx.dom.createElement as newCreateElement\n\n/**\n * Creates a new element

```

```

with the specified [name].\n *\n * The element is initialized
with the specified [init] function.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This
API is moved to another package, use 'kotlinx.dom.createElement' instead.\",\n replaceWith =
ReplaceWith(\"this.createElement(name, init)\",
\"kotlinx.dom.createElement()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic
inline fun Document.createElement(name: String, noinline init: Element.() -> Unit): Element =
this.newCreateElement(name, init)\n\n/**\n * Appends a newly created element with the specified [name] to this
element.\n *\n * The element is initialized with the specified [init] function.\n
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package,
use 'kotlinx.dom.appendChild' instead.\",\n replaceWith = ReplaceWith(\"this.appendChild(name, init)\",
\"kotlinx.dom.appendChild()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic
inline fun Element.appendChild(name:
String, noinline init: Element.() -> Unit): Element = this.newAppendElement(name, init)\n\n\"/>\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport
org.w3c.dom.Element\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.addClass as
newAddClass\nimport kotlinx.dom.hasClass as newHasClass\nimport kotlinx.dom.removeClass as
newRemoveClass\n\n/**\n * Returns true if the element has the given CSS class style in its 'class' attribute
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package,
use 'kotlinx.dom.hasClass' instead.\",\n replaceWith = ReplaceWith(\"this.hasClass(cssClass)\",
\"kotlinx.dom.hasClass()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun
Element.hasClass(cssClass:
String): Boolean = this.newHasClass(cssClass)\n\n/**\n * Adds CSS class to element. Has no effect if all specified
classes are already in class attribute of the element\n *\n * @return true if at least one class has been added\n
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package,
use 'kotlinx.dom.addClass' instead.\",\n replaceWith = ReplaceWith(\"this.addClass(cssClasses)\",
\"kotlinx.dom.addClass()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun
Element.addClass(vararg cssClasses: String): Boolean = this.newAddClass(*cssClasses)\n\n/**\n * Removes all
[cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element\n *\n
* @return true if at least one class has been removed\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n
message = \"This API is moved to another package, use 'kotlinx.dom.removeClass' instead.\",\n replaceWith
= ReplaceWith(\"this.removeClass(cssClasses)\",
\"kotlinx.dom.removeClass()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline
fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses)\n\n\"/>\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.isElement as newIsElement\nimport
kotlinx.dom.isText as newIsText\n\n/**\n * Gets a value indicating whether this node is a TEXT_NODE or a
CDATA_SECTION_NODE.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API
is moved to another package, use 'kotlinx.dom.isText' instead.\",\n replaceWith = ReplaceWith(\"this.isText\",
\"kotlinx.dom.isText()\n)\n)\n@DeprecatedSinceKotlin(warningSince
= \"1.4\", errorSince = \"1.6\")\npublic val Node.isText: Boolean\n inline get() = this.newIsText\n\n/**\n * Gets a
value indicating whether this node is an [Element].\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n
message = \"This API is moved to another package, use 'kotlinx.dom.isElement' instead.\",\n replaceWith =
ReplaceWith(\"this.isElement\", \"kotlinx.dom.isElement()\n)\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.6\")\npublic val Node.isElement: Boolean\n inline get() = this.newIsElement\n\n\"/>\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed

```

```

by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
org.w3c.dom.events\n\npublic fun EventListener(handler: (Event) -> Unit): EventListener =
EventListenerHandler(handler)\n\nprivate class EventListenerHandler(private val handler: (Event) ->
Unit) : EventListener {\n    public override fun handleEvent(event: Event) {\n        handler(event)\n    }\n\n    public
override fun toString(): String = \"EventListenerHandler($handler)\"}\n\n\"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom\n\npublic external interface
ItemArrayLike<out T> {\n    val length: Int\n    fun item(index: Int): T?\n}\n\n/**\n * Returns the view of this
`ItemArrayLike<T>` collection as `List<T>`\n */\n\npublic fun <T> ItemArrayLike<T>.asList(): List<T> = object :
AbstractList<T>() {\n    override val size: Int get() = this@asList.length\n\n    override fun get(index: Int): T = when
(index) {\n        in 0..lastIndex -> this@asList.item(index).unsafeCast<T>()\n        else -> throw
IndexOutOfBoundsException(\"index $index is not in range [0..$lastIndex]\")\n    }\n}\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.appendText as newAppendText\nimport
kotlin.dom.clear as newClear\n\n/** Removes all the children from this node.\n */\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved to another package,
use 'kotlinx.dom.clear' instead.\",\n    replaceWith = ReplaceWith(\"this.clear()\",
\"kotlinx.dom.clear\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\n\npublic inline fun
Node.clear() = this.newClear()\n\n/**\n * Creates text node and append it to the element.\n */\n\n@return this
element\n */\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved to another
package,
use 'kotlinx.dom.appendText' instead.\",\n    replaceWith = ReplaceWith(\"this.appendText(text)\",
\"kotlinx.dom.appendText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\n\ninline fun
Element.appendText(text: String): Element = this.newAppendText(text)\n\n\"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Reinterprets this value
as a value of the [dynamic type](/docs/reference/dynamic-type.html).\n */\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun Any?.asDynamic(): dynamic = this\n\n/**\n * Reinterprets this value as a value of the specified type [T]
without any actual type checking.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Any?.unsafeCast():
@kotlin.internal.NoInfer T = this.asDynamic()\n\n/**\n * Reinterprets this `dynamic` value as a value of
the specified type [T] without any actual type checking.\n */\n\n@kotlin.internal.DynamicExtension\n\n@JsName(\"unsafeCastDynamic\")\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this\n\n/**\n * Allows to iterate this `dynamic`
object in the following cases:\n * - when it has an `iterator` function,\n * - when it is an array\n * - when it is an
instance of [kotlin.collections.Iterable]\n */\n\n@kotlin.internal.DynamicExtension\n\npublic operator fun
dynamic.iterator(): Iterator<dynamic> {\n    val r: Any? = this\n\n    return when {\n        this[\"iterator\"] != null -
->\n            this[\"iterator\"]()\n        isArrayish(r) ->\n            r.unsafeCast<Array<*>>().iterator()\n        else ->\n
            (r as Iterable<*>).iterator()\n    }\n}\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt
file.\n */\n\n// a package is omitted to get declarations directly under the
module\n\n@JsName(\"throwNPE\")\n\ninternal fun throwNPE(message: String) {\n    throw
NullPointerException(message)\n}\n\n@JsName(\"throwCCE\")\n\ninternal fun throwCCE() {\n    throw
ClassCastException(\"Illegal cast\")\n}\n\n@JsName(\"throwISE\")\n\ninternal fun throwISE(message: String) {\n
    throw IllegalStateException(message)\n}\n\n@JsName(\"throwUPAE\")\n\ninternal fun throwUPAE(propertyName:
String) {\n    throw UninitializedPropertyAccessException(\"lateinit property ${propertyName} has not been

```

```

initialized\)\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\npackage kotlin.collections\n\n**\n * Groups elements from the [Grouping] source by key and counts elements\n
in each group.\n *\n * @return a [Map] associating the key of\n
each group with the count of elements in the group.\n *\n * @sample\n
samples.collections.Grouping.groupingByEachCount\n *\n@SinceKotlin("1.1")\npublic actual fun <T, K>\n
Grouping<T, K>.eachCount(): Map<K, Int> =\n fold(0) { acc, _ -> acc + 1 }\n\n/\n**\n * Groups elements from\n
the [Grouping] source by key and sums values provided by the [valueSelector] function for elements in each\n
group.\n *\n * @return a [Map] associating the key of each group with the count of element in the group.\n
*\n@SinceKotlin("1.1")\npublic inline fun <T, K> Grouping<T, K>.eachSumOf(valueSelector: (T) -> Int):\n
Map<K, Int> =\n fold(0) { acc, e -> acc + valueSelector(e) }\n*/\n * Copyright 2010-2018 JetBrains s.r.o.\n
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license\n
that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmName("\nGroupingKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage\n
kotlin.collections\n\n**\n\n * Represents a source of elements with a [keyOf] function, which can be applied to each element to get its key.\n
*\n * A [Grouping] structure serves as an intermediate step in group-and-fold operations:\n * they group elements by\n
their keys and then fold each group with some aggregating operation.\n *\n * It is created by attaching `keySelector`:\n
(T) -> K` function to a source of elements.\n * To get an instance of [Grouping] use one of `groupingBy` extension\n
functions:\n * - [Iterable.groupingBy]\n * - [Sequence.groupingBy]\n * - [Array.groupingBy]\n * -\n
[CharSequence.groupingBy]\n *\n * For the list of group-and-fold operations available, see the [extension\n
functions](#extension-functions) for `Grouping`.\n *\n@SinceKotlin("1.1")\npublic interface Grouping<T, out K>\n
{\n /\n ** Returns an [Iterator] over the elements of the source of this grouping. *\n fun sourceIterator():\n
Iterator<T>\n /\n ** Extracts the key of an [element]. *\n fun keyOf(element: T): K\n}\n\n**\n\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group\n
sequentially,\n * passing the previously accumulated value and the current element as arguments, and stores the\n
results in a new map.\n *\n * The key for each element is provided by the [Grouping.keyOf] function.\n *\n * @param operation function is invoked on each element with the following parameters:\n * - `key` : the key of the\n
group this element belongs to;\n * - `accumulator` : the current value of the accumulator of the group, can be `null`\n
if it's the first `element` encountered in the group;\n * - `element` : the element from the source being aggregated;\n
*\n * - `first` : indicates whether it's the first `element` encountered in the group.\n *\n * @return a [Map] associating\n
the key of each group with the result of aggregation of the group elements.\n *\n * @sample\n
samples.collections.Grouping.aggregateByRadix\n *\n@SinceKotlin("1.1")\npublic inline fun <T, K, R>\n
Grouping<T,\n
K>.aggregate(\n operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R): Map<K, R> {\n return\n
aggregateTo(mutableMapOf<K, R>(), operation)\n}\n\n**\n * Groups elements from the [Grouping] source by key\n
and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and\n
the current element as arguments,\n * and stores the results in the given [destination] map.\n *\n * The key for each\n
element is provided by the [Grouping.keyOf] function.\n *\n * @param operation a function that is invoked on each\n
element with the following parameters:\n * - `key` : the key of the group this element belongs to;\n * -\n
`accumulator` : the current value of the accumulator of the group, can be `null` if it's the first `element` encountered\n
in the group;\n * - `element` : the element from the source being aggregated;\n * - `first` : indicates whether it's the\n
first `element` encountered in the group.\n *\n * If the [destination] map\n
already has a value corresponding to some key,\n * then the elements being aggregated for that key are never\n
considered as `first`.\n *\n * @return the [destination] map associating the key of each group with the result of\n
aggregation of the group elements.\n *\n * @sample samples.collections.Grouping.aggregateByRadixTo\n
*\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T,\n
K>.aggregateTo(\n destination: M,\n operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R): M

```

```

{\n  for (e in this.sourceIterator()) {\n      val key = keyOf(e)\n      val accumulator = destination[key]\n      destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))\n  }\n  return destination\n}\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the\n * elements of each group sequentially,\n * passing the previously accumulated value and the current element as\n * arguments,\n * and stores the results in a new map.\n * An initial value of accumulator is provided by [initialValueSelector]\n * function.\n * @param initialValueSelector a function that provides an initial value of accumulator for each\n * group.\n * It's invoked with parameters:\n * - `key`: the key of the group;\n * - `element`: the first element being\n * encountered in that group.\n * @param operation a function that is invoked on each element with the following\n * parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the\n * accumulator of the group;\n * - `element`: the element from the source being accumulated.\n * @return a [Map]\n * associating the key of each group with the result of accumulating the group elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue\n * @SinceKotlin("1.1")\n * public\n * inline fun <T, K, R> Grouping<T, K>.fold(\n *   initialValueSelector: (key: K, element: T) -> R,\n *   operation: (key: K, accumulator: R, element: T) -> R): Map<K, R> =\n * @Suppress("UNCHECKED_CAST")\n *   aggregate { key, acc, e, first -> operation(key, if (first)\n *     initialValueSelector(key, e) else acc as R, e) }\n * }\n * }\n * Groups elements from the [Grouping] source by key and\n * applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the\n * current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of\n * accumulator is provided by [initialValueSelector] function.\n * @param initialValueSelector a function that\n * provides an initial value of accumulator for each group.\n * It's invoked with parameters:\n * - `key`: the key of the\n * group;\n * - `element`: the first element being encountered in that group.\n * @param destination a [Map] associating the key of each group with the result of accumulating the group\n * elements.\n * If the [destination] map already has\n * a value corresponding to some key, that value is used as an initial value of\n * the accumulator for that group and the\n * [initialValueSelector]\n * function is not called for that group.\n * @param operation a function that is invoked on each element with the\n * following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current\n * value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n * @return the [destination] map associating the key of each group with the result of accumulating the group\n * elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo\n * @SinceKotlin("1.1")\n * public\n * inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n *   destination: M,\n *   initialValueSelector: (key: K, element: T) -> R,\n *   operation: (key: K, accumulator: R, element:\n *   T) -> R): M =\n * @Suppress("UNCHECKED_CAST")\n *   aggregateTo(destination) { key, acc, e, first ->\n *     operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }\n * }\n * }\n * Groups elements from the\n * [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the\n * previously accumulated value and the current element as arguments, and stores the results in a new map.\n * An\n * initial value of accumulator is the same [initialValue] for each group.\n * @param operation a function that is\n * invoked on each element with the following parameters:\n * - `accumulator`: the current value of the accumulator\n * of the group;\n * - `element`: the element from the source being accumulated.\n * @return a [Map] associating\n * the key of each group with the result of accumulating the group elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue\n * @SinceKotlin("1.1")\n * public\n * inline fun <T, K, R> Grouping<T, K>.fold(\n *   initialValue: R,\n *   operation: (accumulator: R, element: T) -> R):\n * Map<K, R> =\n * @Suppress("UNCHECKED_CAST")\n *   aggregate { _, acc, e, first -> operation(if (first)\n *     initialValue else\n *     acc as R, e) }\n * }\n * }\n * Groups elements from the [Grouping] source by key and applies [operation] to the\n * elements of each group sequentially,\n * passing the previously accumulated value and the current element as\n * arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is the same\n * [initialValue] for each group.\n * @param destination a [Map] associating the key of each group with the result of accumulating the group\n * elements.\n * If the [destination] map already has a value corresponding to the key of some\n * group,\n * that value is used as an initial value of the accumulator for that group.\n * @param operation a

```



```

acc + valueSelector(e)}\n\npublic inline fun <T, K> Grouping<T, K>.sumEachByLong(valueSelector: (T) -> Long):
Map<K, Long> =\n    fold(0L) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K, M :
MutableMap<in K, Double>> Grouping<T, K>.sumEachByDoubleTo(destination: M, valueSelector: (T) ->
Double): M =\n    foldTo(destination, 0.0) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K>
Grouping<T, K>.sumEachByDouble(valueSelector: (T) -> Double):
Map<K, Double> =\n    fold(0.0) { acc, e -> acc + valueSelector(e)}\n*/\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\ninternal annotation class JsNativeImplementation(val implementation:
String)\n*/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.js\n\n/**\n * An interface for indexing access to a collection of key-value pairs, where type of
key is [String] and type of value is [Any?][Any].\n */\n\npublic external interface Json {\n    /**\n     * Calls to the
function will be translated to indexing operation (square
brackets) on the receiver with [propertyName] as the argument.\n     * E.g. for next code:\n     * ```kotlin\n
* fun test(j: Json, p: String) = j["prop"] + j.get(p)\n     * ```\n     * will be generated:\n     * ```js\n
* function test(j, p) {\n     *     return j["prop"] + j[p];\n     * }\n     * ```\n     */\n     * operator fun get(propertyName:
String): Any?\n     * Calls of the function will be translated to an assignment of [value] to the receiver
indexed (with square brackets/index operation) with [propertyName].\n     * E.g. for the following code:\n     *
```kotlin\n     * fun test(j: Json, p: String, newValue: Any) {\n     *     j["prop"] = 1\n     *     j.set(p, newValue)\n
* }\n * ```\n * will be generated:\n * ```js\n * function test(j, p, newValue) {\n * j["prop"] =
1;\n * j[p] = newValue;\n * }\n * ```\n */\n * operator fun set(propertyName: String, value:
Any?): Unit\n * Returns a simple JavaScript object (as [Json]) using provided key-value pairs as names
and values of its properties.\n */\n * public fun json(vararg pairs: Pair<String, Any?>): Json {\n * val res: dynamic =
js("{}")\n * for ((name, value) in pairs) {\n * res[name] = value\n * }\n * return res\n * }\n * Adds key-
value pairs from [other] to [this].\n * Returns the original receiver.\n */\n * public fun Json.add(other: Json): Json {\n
* val keys: Array<String> = js("Object").keys(other)\n * if
(other.asDynamic().hasOwnProperty(key)) {\n * this[key] = other[key];\n * }\n * return
this\n * }\n * Exposes the JavaScript [JSON object](https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/JSON) to Kotlin.\n
*/\n\n@Suppress("NOT_DOCUMENTED")\npublic external object JSON {\n public fun stringify(o: Any?):
String\n public fun stringify(o: Any?, replacer: ((key: String,
value: Any?) -> Any?): String)\n public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? =
definedExternally, space: Int): String\n public fun stringify(o: Any?, replacer: ((key: String, value: Any?) ->
Any?)? = definedExternally, space: String): String\n public fun stringify(o: Any?, replacer: Array<String>):
String\n public fun stringify(o: Any?, replacer: Array<String>, space: Int): String\n public fun stringify(o: Any?,
replacer: Array<String>, space: String): String\n\n public fun <T> parse(text: String): T\n public fun <T>
parse(text: String, reviver: ((key: String, value: Any?) -> Any?): T)\n}\n*/\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\nimport
kotlin.internal.InlineOnly\nimport kotlin.js.JsMath as nativeMath\n\n// region =====
Double Math =====\n\n/**\n * Computes the sine of the angle [x]
given in radians.\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n
*/\n\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sin(x: Double): Double =
nativeMath.sin(x)\n\n/**\n * Computes the cosine of the angle [x]
given in radians.\n * Special cases:\n * -
`cos(NaN|+Inf|-Inf)` is `NaN`\n
*/\n\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cos(x: Double):
Double = nativeMath.cos(x)\n\n/**\n * Computes the tangent of the angle [x]
given in radians.\n * Special cases:\n

```

```

* - `tan(NaN|+Inf|-Inf)` is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun tan(x: Double): Double = nativeMath.tan(x)

* Computes the arc sine of the value [x];

* the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.

* Special cases:

* - `asin(x)` is `NaN`, when $abs(x) > 1$ or x is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun asin(x: Double): Double = nativeMath.asin(x)

* Computes the arc cosine of the value [x];

* the returned value is an angle in the range from 0.0 to π radians.

* Special cases:

* - `acos(x)` is `NaN`, when $abs(x) > 1$ or x is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun acos(x: Double): Double = nativeMath.acos(x)

* Computes the arc tangent of the value [x];

* the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.

* Special cases:

* - `atan(NaN)` is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun atan(x: Double): Double = nativeMath.atan(x)

* Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond

* to the rectangular coordinates `(x, y)` by computing the arc tangent of the value y / x ;

* the returned value is an angle in the range from $-\pi$ to π radians.

* Special cases:

* - `atan2(0.0, 0.0)` is 0.0

* - `atan2(0.0, x)` is 0.0 for $x > 0$ and π for $x < 0$

* - `atan2(-0.0, x)` is -0.0 for $x > 0$ and $-\pi$ for $x < 0$

* - `atan2(y, +Inf)` is 0.0 for $0 < y < +Inf$ and -0.0 for $-\text{Inf} < y < 0$

* - `atan2(y, -Inf)` is π for $0 < y < +Inf$ and $-\pi$ for $-\text{Inf} < y < 0$

* - `atan2(y, 0.0)` is $\pi/2$ for $y > 0$ and $-\pi/2$ for $y < 0$

* - `atan2(+Inf, x)` is $\pi/2$ for finite x

* - `atan2(-Inf, x)` is $-\pi/2$ for finite x

* - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun atan2(y: Double, x: Double): Double = nativeMath.atan2(y, x)

* Computes the hyperbolic sine of the value [x].

* Special cases:

* - `sinh(NaN)` is `NaN`

* - `sinh(+Inf)` is $+\text{Inf}$

* - `sinh(-Inf)` is $-\text{Inf}$

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun sinh(x: Double): Double = nativeMath.sinh(x)

* Computes the hyperbolic cosine of the value [x].

* Special cases:

* - `cosh(NaN)` is `NaN`

* - `cosh(+Inf|-Inf)` is $+\text{Inf}$

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)

* Computes the hyperbolic tangent of the value [x].

* Special cases:

* - `tanh(NaN)` is `NaN`

* - `tanh(+Inf)` is 1.0

* - `tanh(-Inf)` is -1.0

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun tanh(x: Double): Double = nativeMath.tanh(x)

* Computes the inverse hyperbolic sine of the value [x].

* The returned value is `y` such that $\sinh(y) == x$.

* Special cases:

* - `asinh(NaN)` is `NaN`

* - `asinh(+Inf)` is $+\text{Inf}$

* - `asinh(-Inf)` is $-\text{Inf}$

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun asinh(x: Double): Double = nativeMath.asinh(x)

* Computes the inverse hyperbolic cosine of the value [x].

* The returned value is positive `y` such that $\cosh(y) == x$.

* Special cases:

* - `acosh(NaN)` is `NaN`

* - `acosh(x)` is `NaN` when $x < 1$

* - `acosh(+Inf)` is $+\text{Inf}$

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun acosh(x: Double): Double = nativeMath.acosh(x)

* Computes the inverse hyperbolic tangent of the value [x].

* The returned value is `y` such that $\tanh(y) == x$.

* Special cases:

* - `tanh(NaN)` is `NaN`

* - `tanh(x)` is `NaN` when $x > 1$ or $x < -1$

* - `tanh(1.0)` is $+\text{Inf}$

* - `tanh(-1.0)` is $-\text{Inf}$

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun atanh(x: Double): Double = nativeMath.atanh(x)

* Computes $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.

* Special cases:

* - returns $+\text{Inf}$ if any of arguments is infinite

* - returns `NaN` if any of arguments is `NaN` and the other is not infinite

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun hypot(x: Double, y: Double): Double = nativeMath.hypot(x, y)

* Computes the positive square root of the value [x].

* Special cases:

* - `sqrt(x)` is `NaN` when $x < 0$ or `x` is `NaN`

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)

* Computes Euler's number `e` raised to the power of the value [x].

* Special cases:

* - `exp(NaN)` is `NaN`

* - `exp(+Inf)` is $+\text{Inf}$

* - `exp(-Inf)` is 0.0

@SinceKotlin("1.2")

@InlineOnly

@public actual inline fun exp(x: Double): Double = nativeMath.exp(x)

* Computes $\exp(x) - 1$.

* This function can be implemented to produce more

```

precise result for [x] near zero.  
Special cases:  
`expm1(NaN)` is `NaN`  
`expm1(+Inf)` is `+Inf`  
`expm1(-Inf)` is `-1.0`  
@see [exp] function.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun expm1(x: Double): Double = nativeMath.expm1(x)

```

Computes the logarithm of the value [x] to the given [base].  
Special cases:  
`log(x, b)` is `NaN` if either `x` or `b` are `NaN`  
`log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0`  
`log(+Inf, +Inf)` is `NaN`  
`log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`  
`log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`  
See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

```

@SinceKotlin("1.2")
public actual
fun log(x: Double, base: Double): Double {
 if (base <= 0.0 || base == 1.0) return Double.NaN
 return nativeMath.log(x) / nativeMath.log(base)
}

```

Computes the natural logarithm (base `E`) of the value [x].  
Special cases:  
`ln(NaN)` is `NaN`  
`ln(x)` is `NaN` when `x < 0.0`  
`ln(+Inf)` is `+Inf`  
`ln(0.0)` is `-Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun ln(x: Double): Double = nativeMath.log(x)

```

Computes the common logarithm (base 10) of the value [x].  
@see [ln] function for special cases.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun log10(x: Double): Double = nativeMath.log10(x)

```

Computes the binary logarithm (base 2) of the value [x].  
@see [ln] function for special cases.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun log2(x: Double): Double = nativeMath.log2(x)

```

Computes `ln(x + 1)`.  
This function can be implemented to produce more precise result for [x] near zero.  
Special cases:  
`ln1p(NaN)` is `NaN`  
`ln1p(x)` is `NaN` where `x < -1.0`  
`ln1p(-1.0)` is `-Inf`  
`ln1p(+Inf)` is `+Inf`  
@see [ln] function  
@see [expm1] function

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun ln1p(x: Double): Double = nativeMath.log1p(x)

```

Rounds the given value [x] to an integer towards positive infinity.  
@return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.  
Special cases:  
`ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun ceil(x: Double): Double = nativeMath.ceil(x)

```

Rounds the given value [x] to an integer towards negative infinity.  
@return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.  
Special cases:  
`floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun floor(x: Double): Double = nativeMath.floor(x)

```

Rounds the given value [x] to an integer towards zero.  
@return the value [x] having its fractional part truncated.  
Special cases:  
`truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun truncate(x: Double): Double = nativeMath.trunc(x)

```

Rounds the given value [x] towards the closest integer with ties rounded towards even integer.  
Special cases:  
`round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

```

@SinceKotlin("1.2")
public actual
fun round(x: Double): Double {
 if (x % 0.5 != 0.0) {
 return nativeMath.round(x)
 }
 val floor = floor(x)
 return if (floor % 2 == 0.0) floor else ceil(x)
}

```

Returns the absolute value of the given value [x].  
Special cases:  
`abs(NaN)` is `NaN`  
@see absoluteValue extension property for [Double]

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun abs(x: Double): Double = nativeMath.abs(x)

```

Returns the sign of the given value [x]:  
`-1.0` if the value is negative,  
zero if the value is zero,  
`1.0` if the value is positive  
Special case:  
`sign(NaN)` is `NaN`

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun sign(x: Double): Double = nativeMath.sign(x)

```

Returns the smaller of two values.  
If either value is `NaN`, then the result is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun min(a: Double, b: Double): Double = nativeMath.min(a, b)

```

Returns the greater of two values.  
If either value is `NaN`, then the result is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public actual
inline fun max(a: Double, b: Double): Double =

```

```

nativeMath.max(a, b)\n\n// extensions\n\n**\n * Raises this value to the power [x].\n * Special cases:\n * -
`b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is `NaN` for `x !=
0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not
an integer\n *^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic
actual inline fun Double.pow(x: Double): Double = nativeMath.pow(this, x)\n\n**\n * Raises this value to the
integer power [n].\n * See the other overload of [pow] for details.\n
*^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Double.pow(n: Int): Double =
nativeMath.pow(this, n.toDouble())\n\n**\n * Returns the absolute value of this value.\n * Special cases:\n * -
`NaN.absoluteValue` is `NaN`\n * * @see abs function\n *^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual
inline val Double.absoluteValue: Double get() = nativeMath.abs(this)\n\n**\n * Returns the sign of this value:\n *
- `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n * * Special
case:\n * - `NaN.sign` is `NaN`\n *^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Double.sign:
Double get() = nativeMath.sign(this)\n\n**\n * Returns this value with the sign bit same as
of the [sign] value.\n *^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Double.withSign(sign: Int):
Double = this.withSign(sign.toDouble())\n\n**\n * Returns the ulp (unit in the last place) of this value.\n * An
ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.\n * * Special
Cases:\n * - `NaN.ulp` is `NaN`\n * - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf`\n * - `0.0.ulp` is
`Double.MIN_VALUE`\n *^@\n@SinceKotlin("1.2")\npublic actual val Double.ulp: Double get() = when {\n this
< 0 -> (-this).ulp\n this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n this ==
Double.MAX_VALUE -> this - this.nextDown()\n else -> this.nextUp() - this\n}\n\n**\n * Returns the [Double]
value nearest to this value in direction of positive infinity.\n *^@\n@SinceKotlin("1.2")\npublic actual fun
Double.nextUp(): Double = when {\n this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n this == 0.0
-> Double.MIN_VALUE\n else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)\n}\n\n**\n *
Returns the [Double] value nearest to this value in direction of negative infinity.\n
*^@\n@SinceKotlin("1.2")\npublic actual fun Double.nextDown(): Double = when {\n this.isNaN() || this ==
Double.NEGATIVE_INFINITY -> this\n this == 0.0 -> -Double.MIN_VALUE\n else ->
Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)\n}\n\n**\n * Returns the [Double] value nearest to this
value in direction from this value towards the value [to].\n * Special cases:\n * - `x.nextTowards(y)` is `NaN` if
either `x` or `y` are `NaN`\n * - `x.nextTowards(x) == x`\n *^@\n@SinceKotlin("1.2")\npublic actual fun
Double.nextTowards(to: Double): Double = when {\n this.isNaN() || to.isNaN() -> Double.NaN\n to == this ->
to\n to > this -> this.nextUp()\n else /* to < this */ -> this.nextDown()\n}\n\n**\n * Rounds this [Double]
value to the nearest integer and
converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * -
`x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE`
when `x < Int.MIN_VALUE`\n * * @throws IllegalArgumentException when this value is `NaN`\n
*^@\n@SinceKotlin("1.2")\npublic actual fun Double.roundToInt(): Int = when {\n isNaN() -> throw
IllegalArgumentException("Cannot round NaN value.")\n this > Int.MAX_VALUE -> Int.MAX_VALUE\n
this < Int.MIN_VALUE -> Int.MIN_VALUE\n else -> nativeMath.round(this).toInt()\n}\n\n**\n * Rounds this
[Double] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive
infinity.\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x >
Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n * *
@throws IllegalArgumentException when this value is `NaN`\n *^@\n@SinceKotlin("1.2")\npublic
actual fun Double.roundToLong(): Long = when {\n isNaN() -> throw IllegalArgumentException("Cannot round
NaN value.")\n this > Long.MAX_VALUE -> Long.MAX_VALUE\n this < Long.MIN_VALUE ->
Long.MIN_VALUE\n else -> nativeMath.round(this).toLong()\n}\n\n// endregion\n\n// region
===== Float Math =====\n\n**\n * Computes the
sine of the angle [x] given in radians.\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n
*^@\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sin(x: Float): Float =

```

`nativeMath.sin(x.toDouble()).toFloat()`  
 Computes the cosine of the angle [x] given in radians.  
 Special cases: `cos(NaN|+Inf|-Inf)` is `NaN`

`cos(x: Float): Float = nativeMath.cos(x.toDouble()).toFloat()`  
 Computes the tangent of the angle [x] given in radians.  
 Special cases: `tan(NaN|+Inf|-Inf)` is `NaN`

`tan(x: Float): Float = nativeMath.tan(x.toDouble()).toFloat()`  
 Computes the arc sine of the value [x];  
 the returned value is an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.  
 Special cases: `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`

`asin(x: Float): Float = nativeMath.asin(x.toDouble()).toFloat()`  
 Computes the arc cosine of the value [x];  
 the returned value is an angle in the range from  $0.0$  to  $\pi$  radians.  
 Special cases: `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`

`acos(x: Float): Float = nativeMath.acos(x.toDouble()).toFloat()`  
 Computes the arc tangent of the value [x];  
 the returned value is an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.  
 Special cases: `atan(NaN)` is `NaN`

`atan(x: Float): Float = nativeMath.atan(x.toDouble()).toFloat()`  
 Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value `y / x`;  
 the returned value is an angle in the range from  $-\pi$  to  $\pi$  radians.  
 Special cases: `atan2(0.0, 0.0)` is `0.0`  
`atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`  
`atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`  
`atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for `-Inf < y < 0`  
`atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`  
`atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`  
`atan2(+Inf, x)` is `PI/2` for finite `x`  
`atan2(-Inf, x)` is `-PI/2` for finite `x`  
`atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`

`atan2(y: Float, x: Float): Float = nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()`  
 Computes the hyperbolic sine of the value [x].  
 Special cases: `sinh(NaN)` is `NaN`  
`sinh(+Inf)` is `+Inf`  
`sinh(-Inf)` is `-Inf`

`sinh(x: Float): Float = nativeMath.sinh(x.toDouble()).toFloat()`  
 Computes the hyperbolic cosine of the value [x].  
 Special cases: `cosh(NaN)` is `NaN`  
`cosh(+Inf|-Inf)` is `+Inf`

`cosh(x: Float): Float = nativeMath.cosh(x.toDouble()).toFloat()`  
 Computes the hyperbolic tangent of the value [x].  
 Special cases: `tanh(NaN)` is `NaN`  
`tanh(+Inf)` is `1.0`  
`tanh(-Inf)` is `-1.0`

`tanh(x: Float): Float = nativeMath.tanh(x.toDouble()).toFloat()`  
 Computes the inverse hyperbolic sine of the value [x].  
 The returned value is `y` such that `sinh(y) == x`.  
 Special cases: `asinh(NaN)` is `NaN`  
`asinh(+Inf)` is `+Inf`  
`asinh(-Inf)` is `-Inf`

`asinh(x: Float): Float = nativeMath.asinh(x.toDouble()).toFloat()`  
 Computes the inverse hyperbolic cosine of the value [x].  
 The returned value is positive `y` such that `cosh(y) == x`.  
 Special cases: `acosh(NaN)` is `NaN`  
`acosh(x)` is `NaN` when `x < 1`  
`acosh(+Inf)` is `+Inf`

`acosh(x: Float): Float = nativeMath.acosh(x.toDouble()).toFloat()`  
 Computes the inverse hyperbolic tangent of the value [x].  
 The returned value is `y` such that `tanh(y) == x`.  
 Special cases: `atanh(NaN)` is `NaN`  
`atanh(x)` is `NaN`  
 when `x > 1` or `x < -1`  
`atanh(1.0)` is `+Inf`  
`atanh(-1.0)` is `-Inf`

`atanh(x: Float): Float = nativeMath.atanh(x.toDouble()).toFloat()`  
 Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.  
 Special cases: returns `+Inf` if any of arguments is infinite  
 returns `NaN` if any of arguments is `NaN` and the other is not infinite

`hypot(x: Float, y: Float): Float = nativeMath.hypot(x.toDouble(), y.toDouble()).toFloat()`  
 Computes the

positive square root of the value [x].  
 \* Special cases:  
 $\sqrt{x}$  is NaN when  $x < 0$  or  $x$  is NaN  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun sqrt(x: Float): Float =  
 nativeMath.sqrt(x.toDouble()).toFloat()  
 \* Computes Euler's number  $e$  raised to the power of the value [x].  
 \* Special cases:  
 $\exp(\text{NaN})$  is NaN  
 $\exp(+\text{Inf})$  is  $+\text{Inf}$   
 $\exp(-\text{Inf})$  is  $0.0$   
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun exp(x: Float): Float = nativeMath.exp(x.toDouble()).toFloat()  
 \* Computes  $\exp(x) - 1$ .  
 \* This function can be implemented to produce more precise result for [x] near zero.  
 \* Special cases:  
 $\expm1(\text{NaN})$  is NaN  
 $\expm1(+\text{Inf})$  is  $+\text{Inf}$   
 $\expm1(-\text{Inf})$  is  $-1.0$   
 \* @see [exp] function.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun expm1(x: Float): Float =  
 nativeMath.expm1(x.toDouble()).toFloat()  
 \* Computes the logarithm of the value [x] to the given [base].  
 \* Special cases:  
 $\log(x, b)$  is NaN if either  $x$  or  $b$  are NaN  
 $\log(x, b)$  is NaN when  $x < 0$  or  $b \leq 0$  or  $b = 1.0$   
 $\log(+\text{Inf}, +\text{Inf})$  is NaN  
 $\log(+\text{Inf}, b)$  is  $+\text{Inf}$  for  $b > 1$  and  $-\text{Inf}$  for  $b < 1$   
 $\log(0.0, b)$  is  $-\text{Inf}$  for  $b > 1$  and  $+\text{Inf}$  for  $b > 1$   
 \* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun log(x: Float, base: Float): Float =  
 log(x.toDouble(), base.toDouble()).toFloat()  
 \* Computes the natural logarithm (base  $E$ ) of the value [x].  
 \* Special cases:  
 $\ln(\text{NaN})$  is NaN  
 $\ln(x)$  is NaN when  $x < 0.0$   
 $\ln(+\text{Inf})$  is  $+\text{Inf}$   
 $\ln(0.0)$  is  $-\text{Inf}$   
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun ln(x: Float): Float =  
 nativeMath.log(x.toDouble()).toFloat()  
 \* Computes the common logarithm (base 10) of the value [x].  
 \* @see [ln] function for special cases.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun log10(x: Float): Float = nativeMath.log10(x.toDouble()).toFloat()  
 \* Computes the binary logarithm (base 2) of the value [x].  
 \* @see [ln] function for special cases.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun log2(x: Float): Float = nativeMath.log2(x.toDouble()).toFloat()  
 \* Computes  $\ln(a + 1)$ .  
 \* This function can be implemented to produce more precise result for [x] near zero.  
 \* Special cases:  
 $\ln1p(\text{NaN})$  is NaN  
 $\ln1p(x)$  is NaN where  $x < -1.0$   
 $\ln1p(-1.0)$  is  $-\text{Inf}$   
 $\ln1p(+\text{Inf})$  is  $+\text{Inf}$   
 \* @see [ln] function  
 \* @see [expm1] function  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun ln1p(x: Float): Float = nativeMath.log1p(x.toDouble()).toFloat()  
 \* Rounds the given value [x] to an integer towards positive infinity.  
 \* @return the smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.  
 \* Special cases:  
 $\text{ceil}(x)$  is  $x$  where  $x$  is NaN or  $+\text{Inf}$  or  $-\text{Inf}$  or already a mathematical integer.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun ceil(x: Float): Float = nativeMath.ceil(x.toDouble()).toFloat()  
 \* Rounds the given value [x] to an integer towards negative infinity.  
 \* @return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.  
 \* Special cases:  
 $\text{floor}(x)$  is  $x$  where  $x$  is NaN or  $+\text{Inf}$  or  $-\text{Inf}$  or already a mathematical integer.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun floor(x: Float): Float =  
 nativeMath.floor(x.toDouble()).toFloat()  
 \* Rounds the given value [x] to an integer towards zero.  
 \* @return the value [x] having its fractional part truncated.  
 \* Special cases:  
 $\text{truncate}(x)$  is  $x$  where  $x$  is NaN or  $+\text{Inf}$  or  $-\text{Inf}$  or already a mathematical integer.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun truncate(x: Float): Float = truncate(x.toDouble()).toFloat()  
 \* Rounds the given value [x] towards the closest integer with ties rounded towards even integer.  
 \* Special cases:  
 $\text{round}(x)$  is  $x$  where  $x$  is NaN or  $+\text{Inf}$  or  $-\text{Inf}$  or already a mathematical integer.  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun round(x: Float): Float =  
 round(x.toDouble()).toFloat()  
 \* Returns the absolute value of the given value [x].  
 \* Special cases:  
 $\text{abs}(\text{NaN})$  is NaN  
 \* @see absoluteValue extension property for [Float]  
 \* Since Kotlin("1.2")  
 @InlineOnly  
 public actual inline fun abs(x: Float): Float =  
 nativeMath.abs(x.toDouble()).toFloat()  
 \* Returns the sign of the given value [x]:  
 $-1.0$  if the value is negative,  
 $0.0$  if the value is zero,  
 $1.0$  if the value is positive  
 \* Special case:  
 $\text{sign}(\text{NaN})$  is NaN

is `NaN`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sign(x: Float): Float =
 nativeMath.sign(x.toDouble()).toFloat()

```

Returns the smaller of two values. If either value is `NaN`, then the result is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun min(a: Float, b: Float): Float = nativeMath.min(a, b)

```

Returns the greater of two values. If either value is `NaN`, then the result is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun max(a: Float, b: Float): Float = nativeMath.max(a, b)

```

Raises this value to the power `[x]`. Special cases: `b.pow(0.0)` is `1.0`, `b.pow(1.0) == b`, `b.pow(NaN)` is `NaN`, `NaN.pow(x)` is `NaN` for `x != 0.0`, `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`, `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.pow(x: Float): Float = nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()

```

Raises this value to the integer power `[n]`. See the other overload of `[pow]` for details.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.pow(n: Int): Float
 = nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()

```

Returns the absolute value of this value. Special cases: `NaN.absoluteValue` is `NaN`. @see `abs` function

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline val Float.absoluteValue: Float get() =
 nativeMath.abs(this.toDouble()).toFloat()

```

Returns the sign of this value: `-1.0` if the value is negative, `0` if the value is zero, `1.0` if the value is positive. Special case: `NaN.sign` is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline val Float.sign: Float get() =
 nativeMath.sign(this.toDouble()).toFloat()

```

Returns this value with the sign bit same as of the `[sign]` value. If `[sign]` is `NaN` the sign of the result is undefined.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.withSign(sign: Float): Float =
 this.toDouble().withSign(sign.toDouble()).toFloat()

```

Returns this value with the sign bit same as of the `[sign]` value.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.withSign(sign: Int): Float = this.toDouble().withSign(sign.toDouble()).toFloat()

```

Rounds this `[Float]` value to the nearest integer and converts the result to `[Int]`. Ties are rounded towards positive infinity. Special cases: `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`, `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`. @throws `IllegalArgumentException` when this value is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.roundToInt():
 Int = toDouble().roundToInt()

```

Rounds this `[Float]` value to the nearest integer and converts the result to `[Long]`. Ties are rounded towards positive infinity. Special cases: `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`, `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`. @throws `IllegalArgumentException` when this value is `NaN`.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun Float.roundToLong(): Long =
 toDouble().roundToLong()

```

endregion  
region ===== Integer Math  
=====

Returns the absolute value of the given value `[n]`. Special cases: `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow. @see `absoluteValue` extension property for `[Int]`. TODO: remove manual 'or' when KT-19290 is fixed.

```

@SinceKotlin("1.2")
public
 actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n

```

Returns the smaller of two values.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun min(a: Int, b: Int): Int =
 nativeMath.min(a, b)

```

Returns the greater of two values.

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline fun max(a: Int, b: Int): Int = nativeMath.max(a,
 b)

```

Returns the absolute value of this value. Special cases: `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow. @see `abs` function

```

@SinceKotlin("1.2")
@InlineOnly
public
 actual inline val Int.absoluteValue: Int get() = abs(this)

```

Returns the sign of this value: `-1` if the value is negative, `0` if the value is zero, `1` if the value is positive.

```

@SinceKotlin("1.2")
public
 actual val Int.sign:
 Int get() = when {
 this < 0 -> -1
 this > 0 -> 1
 else -> 0
 }

```

Returns the absolute value of the



given value [n].  
Special cases: `-abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow  
@see `absoluteValue` extension property for [Long]  
@SinceKotlin("1.2")  
public actual fun `abs(n: Long): Long` = if (n < 0) -n else n  
Returns the smaller of two values.  
@SinceKotlin("1.2")  
@Suppress("NOTHING\_TO\_INLINE")  
public actual inline fun `min(a: Long, b: Long): Long`:  
Long = if (a <= b) a else b  
Returns the greater of two values.  
@SinceKotlin("1.2")  
@Suppress("NOTHING\_TO\_INLINE")  
public actual inline fun `max(a: Long, b: Long): Long`:  
Long = if (a >= b) a else b  
Returns the absolute value of this value.  
Special cases: `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow  
@see `abs` function  
@SinceKotlin("1.2")  
@InlineOnly  
public actual inline val `Long.absoluteValue: Long` get() = `abs(this)`  
Returns the sign of this value: `-1` if the value is negative, `0` if the value is zero, `1` if the value is positive  
@SinceKotlin("1.2")  
public actual val `Long.sign: Int` get() = when {  
this < 0 -> -1  
this > 0 -> 1  
else -> 0  
} // endregion  
"/\*\*  
Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.  
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  
package kotlin  
Returns `true` if the specified number is a Not-a-Number (NaN) value, `false` otherwise.  
public actual fun `Double.isNaN(): Boolean` = this != this  
Returns `true` if the specified number is a Not-a-Number (NaN) value, `false` otherwise.  
public actual fun `Float.isNaN(): Boolean` = this != this  
Returns `true` if this value is infinitely large in magnitude.  
public actual fun `Double.isInfinite(): Boolean` = this == Double.POSITIVE\_INFINITY || this == Double.NEGATIVE\_INFINITY  
Returns `true` if this value is infinitely large in magnitude.  
public actual fun `Float.isInfinite(): Boolean` = this == Float.POSITIVE\_INFINITY || this == Float.NEGATIVE\_INFINITY  
Returns `true` if the argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).  
public actual fun `Double.isFinite(): Boolean` = !isInfinite() && !isNaN()  
Returns `true` if the argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).  
public actual fun `Float.isFinite(): Boolean` = !isInfinite() && !isNaN()  
Counts the number of set bits in the binary representation of this [Int] number.  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
public actual fun `Int.countOneBits(): Int` {  
// Hacker's Delight 5-1 algorithm  
var v = this  
v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)  
v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)  
v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)  
v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)  
v = (v and 0x0000FFFF) + (v.ushr(16))  
return v  
} // endregion  
"/\*\*  
Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
@kotlin.internal.InlineOnly  
public actual inline fun `Int.countLeadingZeroBits(): Int` = JsMath.clz32(this)  
Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
public actual fun `Int.countTrailingZeroBits(): Int` =  
Int.SIZE\_BITS - (this or -this).inv().countLeadingZeroBits()  
Returns a number having a single bit set in the position of the most significant set bit of this [Int] number, or zero, if this number is zero.  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
public actual fun `Int.takeHighestOneBit(): Int` =  
if (this == 0) 0 else 1.shl(Int.SIZE\_BITS - 1 - countLeadingZeroBits())  
Returns a number having a single bit set in the position of the least significant set bit of this [Int] number, or zero, if this number is zero.  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
public actual fun `Int.takeLowestOneBit(): Int` =  
// Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit  
this and -

this

`Int.rotateLeft(bitCount: Int): Int` = `shl(bitCount)` or `ushr(Int.SIZE_BITS - bitCount)`

\* Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits. The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.

\* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count: `number.rotateLeft(-n) == number.rotateRight(n)`

\* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally `number.rotateLeft(n) == number.rotateLeft(n % 32)`

`@SinceKotlin("1.6") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Int.rotateRight(bitCount: Int): Int` = `shl(Int.SIZE_BITS - bitCount)` or `ushr(bitCount)`

\* Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits. The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.

\* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count: `number.rotateRight(-n) == number.rotateLeft(n)`

\* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally `number.rotateRight(n) == number.rotateRight(n % 32)`

`@SinceKotlin("1.6") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Int.rotateRight(bitCount: Int): Int` = `shl(Int.SIZE_BITS - bitCount)` or `ushr(bitCount)`

\* Counts the number of set bits in the binary representation of this [Long] number.

`@SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.countOneBits(): Int` = `high.countOneBits() + low.countOneBits()`

\* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.

`@SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.countLeadingZeroBits(): Int` = `when (val high = this.high) { 0 -> Int.SIZE_BITS + low.countLeadingZeroBits() else -> high.countLeadingZeroBits() }`

\* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.

`@SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.countTrailingZeroBits(): Int` = `when (val low = this.low) { 0 -> Int.SIZE_BITS + high.countTrailingZeroBits() else -> low.countTrailingZeroBits() }`

\* Returns a number having a single bit set in the position of the most significant set bit of this [Long] number, or zero, if this number is zero.

`@SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.takeHighestOneBit(): Long` = `when (val high = this.high) { 0 -> Long(low.takeHighestOneBit(), 0) else -> Long(0, high.takeHighestOneBit()) }`

\* Returns a number having a single bit set in the position of the least significant set bit of this [Long] number, or zero, if this number is zero.

`@SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.takeLowestOneBit(): Long` = `when (val low = this.low) { 0 -> Long(0, high.takeLowestOneBit()) else -> Long(low.takeLowestOneBit(), 0) }`

\* Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits. The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.

\* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count: `number.rotateLeft(-n) == number.rotateRight(n)`

\* Rotating by a multiple of [Long.SIZE\_BITS] (64) returns the same number, or more generally `number.rotateLeft(n) == number.rotateLeft(n % 64)`

`@SinceKotlin("1.6") @WasExperimental(ExperimentalStdlibApi::class) public actual fun`

`Long.rotateLeft(bitCount: Int): Long` = `if ((bitCount and 31) != 0) { val low = this.low val high = this.high val newLow = low.shl(bitCount) or high.ushr(-bitCount) val newHigh = high.shl(bitCount) or low.ushr(-bitCount) return if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(newHigh, newLow) } else { return if ((bitCount and 32) == 0) this else Long(high, low) }`

\* Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits. The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.

\* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count: `number.rotateRight(-n) == number.rotateLeft(n)`

\* Rotating by a

```

multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally
number.rotateRight(n) ==
number.rotateRight(n % 64)

*\/n@SinceKotlin("1.6")\/n@WasExperimental(ExperimentalStdlibApi::class)\/n@kotlin.internal.InlineOnly\/npublic
c actual inline fun Long.rotateRight(bitCount: Int): Long = rotateLeft(-bitCount)\/n", "/*\/n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n *\/n\/npackage kotlin.js\/n\/nimport
kotlin.internal.LowPriorityInOverloadResolution\/n\/n**\/n * Exposes the JavaScript [Promise
object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise) to Kotlin.\/n
*\/n@Suppress("NOT_DOCUMENTED")\/npublic open external class Promise<out T>(executor: (resolve: (T) ->
Unit, reject: (Throwable) -> Unit) -> Unit) {\/n @LowPriorityInOverloadResolution\/n public open fun <S>
then(onFulfilled: ((T) -> S)?): Promise<S>\/n\/n @LowPriorityInOverloadResolution\/n public open fun <S>
then(onFulfilled: ((T) -> S)?, onRejected: ((Throwable) -> S)?): Promise<S>\/n\/n public open fun <S>
catch(onRejected: (Throwable) -> S): Promise<S>\/n\/n public open fun finally(onFinally: () -> Unit):
Promise<T>\/n\/n companion object {\/n public fun <S> all(promise: Array<out Promise<S>>):
Promise<Array<out S>>\/n\/n public fun <S> race(promise: Array<out Promise<S>>): Promise<S>\/n\/n
public fun reject(e: Throwable): Promise<Nothing>\/n\/n public fun <S> resolve(e: S): Promise<S>\/n public
fun <S> resolve(e: Promise<S>): Promise<S>\/n }\/n}\/n\/n// It's workaround for KT-19672 since we can fix it
properly until KT-11265 isn't fixed.\/ninline
fun <T, S> Promise<Promise<T>>.then(\/n noinline onFulfilled: ((T) -> S)?): Promise<S> {\/n return
this.unsafeCast<Promise<T>>().then(onFulfilled)\/n}\/n}\/ninline fun <T, S> Promise<Promise<T>>.then(\/n noinline
onFulfilled: ((T) -> S)?,\/n noinline onRejected: ((Throwable) -> S)?): Promise<S> {\/n return
this.unsafeCast<Promise<T>>().then(onFulfilled, onRejected)\/n}\/n}\/n", "/*\/n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\/n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\/n *\/n\/npackage kotlin.random\/n\/nimport
kotlin.math.pow\/n\/ninternal actual fun defaultPlatformRandom(): Random =\/n Random(js("Math.random() *
Math.pow(2, 32)) | 0").unsafeCast<Int>())\/n\/nprivate val INV_2_26: Double = 2.0.pow(-26)\/nprivate val
INV_2_53: Double = 2.0.pow(-53)\/ninternal actual fun doubleFromParts(hi26: Int, low27: Int): Double =\/n hi26 *
INV_2_26 + low27 * INV_2_53", "/*\/n
* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n *\/n\/npackage
kotlin.reflect\/n\/nimport findAssociatedObject\/n\/n**\/n * The experimental marker for associated objects API.\/n *\/n
* Any usage of a declaration annotated with `@ExperimentalAssociatedObjects` must be accepted either by\/n *
annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalAssociatedObjects::class)`,\/n * or by
using the compiler argument `-opt-in=kotlin.reflect.ExperimentalAssociatedObjects`.\/n *\/n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\/n@Retention(value = AnnotationRetention.BINARY)\/npublic annotation class
ExperimentalAssociatedObjects\/n\/n**\/n * Makes the annotated annotation class an associated object key.\/n *\/n
* An associated object key annotation should have single [KClass] parameter.\/n * When applied to a class with
reference
to an object declaration as an argument, it binds\/n * the object to the class, making this binding discoverable at
runtime using [findAssociatedObject].\/n
*\/n@ExperimentalAssociatedObjects\/n@Retention(AnnotationRetention.BINARY)\/n@Target(AnnotationTarget.A
NNOTATION_CLASS)\/npublic annotation class AssociatedObjectKey\/n\/n**\/n * If [T] is an
@[AssociatedObjectKey]-annotated annotation class and [this] class is annotated with @[T](`S::class`),\/n * returns
object `S`.\/n * Otherwise returns `null`.\/n *\/n@ExperimentalAssociatedObjects\/npublic inline fun <reified T :
Annotation> KClass<*>.findAssociatedObject(): Any? =\/n this.findAssociatedObject(T::class)", "/*\/n * Copyright
2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n *\/n\/npackage kotlin.js\/n\/nimport
getKClass\/n\/nimport kotlin.reflect.KClass\/n\/nimport kotlin.reflect.js.internal.KClassImpl\/n\/n**\/n

```

\* Represents the constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a constructor reference.

```

external interface JsClass<T : Any> {
 /** Returns the unqualified name of the class represented by this instance.
 * val name: String
 }

 /** Obtains a constructor reference for the given `KClass`.
 * nval <T : Any> KClass<T>.js: JsClass<T>
 * get() = (this as KClassImpl<T>).jClass

 /** Obtains a `KClass` instance for the given constructor reference.
 * nval <T : Any> JsClass<T>.kotlin: KClass<T>
 * get() = getKClass(this)

 /** Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * npackage kotlin.reflect.js.internal
 * nimport kotlin.reflect.*
 * ninternal abstract class KClassImpl<T : Any> {
 * internal open val jClass: JsClass<T>
 * : KClass<T>
 * override val qualifiedName: String?
 * get() = TODO()
 * override fun equals(other: Any?): Boolean {
 * return other is KClassImpl<*> && jClass == other.jClass
 * }
 * // TODO: use FQN
 * override fun hashCode(): Int = simpleName?.hashCode() ?: 0
 * override fun toString(): String {
 * // TODO: use FQN
 * return "class $simpleName"
 * }
 * ninternal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) : KClassImpl<T>(jClass) {
 * override val simpleName: String? = jClass.asDynamic().$metadata?.simpleName.unsafeCast<String?>()
 * override fun isInstance(value: Any?): Boolean {
 * return jsIsType(value, jClass)
 * }
 * ninternal class PrimitiveKClassImpl<T : Any>(jClass: JsClass<T>, private val givenSimpleName: String, private val isInstanceFunction: (Any?) -> Boolean) : KClassImpl<T>(jClass) {
 * override fun equals(other: Any?): Boolean {
 * if (other is PrimitiveKClassImpl<*>) return false
 * return super.equals(other) && givenSimpleName == other.givenSimpleName
 * }
 * override val simpleName: String? get() = givenSimpleName
 * override fun isInstance(value: Any?): Boolean {
 * return isInstanceFunction(value)
 * }
 * ninternal object NothingKClassImpl : KClassImpl<Nothing>(js("Object")) {
 * override val simpleName: String = "Nothing"
 * override fun isInstance(value: Any?): Boolean = false
 * override val jClass: JsClass<Nothing>
 * get() = throw UnsupportedOperationException("There's no native JS class for Nothing type")
 * override fun equals(other: Any?): Boolean = other === this
 * override fun hashCode(): Int = 0
 * }
 * ninternal class ErrorKClass : KClass<Nothing> {
 * override val simpleName: String? get() = error("Unknown simpleName for ErrorKClass")
 * override val qualifiedName: String? get() = error("Unknown qualifiedName for ErrorKClass")
 * override fun isInstance(value: Any?): Boolean = error("Can's check isInstance on ErrorKClass")
 * override fun equals(other: Any?): Boolean = other === this
 * override fun hashCode(): Int = 0
 * }
 * }
 * }
 * }

 /** Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * npackage kotlin.reflect
 * ninternal actual inline val KClass<*>.qualifiedOrSimpleName: String?
 * get() = simpleName

 /** Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * npackage // a package is omitted to get declarations directly under the module
 * n// TODO: Remove once JsReflectionAPICallChecker supports more reflection
 * types
 * n@file:Suppress("Unsupported")
 * nimport kotlin.reflect.*
 * nimport kotlin.reflect.js.internal.*
 * n@JsName("createKType")
 * ninternal fun createKType(
 * classifier: KClassifier,
 * arguments: Array<KTypeProjection>,
 * isMarkedNullable: Boolean) =
 * KTypeImpl(classifier, arguments.asList(), isMarkedNullable)
 * n@JsName("createDynamicKType")
 * ninternal fun createDynamicKType(): KType = DynamicKType
 * n@JsName("markKTypeNullable")
 * ninternal fun markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments, true)
 * n@JsName("createKTypeParameter")
 * ninternal fun createKTypeParameter(
 * name: String,
 * upperBounds: Array<KType>,
 * variance: String) : KTypeParameter {
 * val kVariance = when (variance) {
 * "in" -> KVariance.IN
 * "out" -> KVariance.OUT
 * else -> KVariance.INVARIANT
 * }
 * return KTypeParameterImpl(name, upperBounds.asList(), kVariance, false)
 * }
 * n@JsName("getStarKTypeProjection")
 * ninternal fun getStarKTypeProjection(): KTypeProjection =

```

```

KTypeProjection.STAR\n\n@jsName("createCovariantKTypeProjection")\ninternal fun
createCovariantKTypeProjection(type: KType): KTypeProjection =\n
KTypeProjection.covariant(type)\n\n@jsName("createInvariantKTypeProjection")\ninternal fun
createInvariantKTypeProjection(type: KType): KTypeProjection =\n
KTypeProjection.invariant(type)\n\n@jsName("createContravariantKTypeProjection")\ninternal fun
createContravariantKTypeProjection(type: KType): KTypeProjection =\n
KTypeProjection.contravariant(type)\n\n/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal class
KTypeImpl(\n override val classifier: KClassifier,\n override val arguments: List<KTypeProjection>,\n override val isMarkedNullable: Boolean\n) : KType {\n override fun equals(other: Any?): Boolean =\n other
is KTypeImpl &&\n classifier
== other.classifier && arguments == other.arguments && isMarkedNullable == other.isMarkedNullable\n\n override fun hashCode(): Int =\n (classifier.hashCode() * 31 + arguments.hashCode()) * 31 +
isMarkedNullable.hashCode()\n\n override fun toString(): String {\n val kClass = (classifier as? KClass<*>)\n val classifierName = when {\n kClass == null -> classifier.toString()\n kClass.simpleName != null
-> kClass.simpleName\n else -> "(non-denotable type)"\n }\n val args =\n if
(arguments.isEmpty()) ""\n else arguments.joinToString(", ", "<", ">")\n val nullable = if
(isMarkedNullable) "?" else ""\n return classifierName + args + nullable\n }\n}\n\ninternal object
DynamicKType : KType {\n override val classifier: KClassifier? = null\n override val arguments:
List<KTypeProjection> = emptyList()\n override val isMarkedNullable: Boolean = false\n\n override fun toString(): String = "dynamic"\n}\n\n/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal
data class KTypeParameterImpl(\n override val name: String,\n override val upperBounds: List<KType>,\n override val variance: KVariance,\n override val isReified: Boolean\n) : KTypeParameter {\n override fun
toString(): String = name\n}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport
kotlin.js.JsClass\n\n@jsName("PrimitiveClasses")\ninternal object PrimitiveClasses {\n\n @JsName("anyClass")\n val anyClass = PrimitiveKClassImpl(js("Object").unsafeCast<JsClass<Any>>(),
"Any", { it is Any })\n\n @JsName("numberClass")\n val numberClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Number>>(), "Number", { it is Number })\n\n @JsName("nothingClass")\n val nothingClass = NothingKClassImpl\n\n @JsName("booleanClass")\n val
booleanClass = PrimitiveKClassImpl(js("Boolean").unsafeCast<JsClass<Boolean>>(), "Boolean", { it is Boolean
})\n\n @JsName("byteClass")\n val byteClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Byte>>(), "Byte", { it is Byte })\n\n @JsName("shortClass")\n val shortClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Short>>(),
"Short", { it is Short })\n\n @JsName("intClass")\n val intClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(), "Int", { it is Int })\n\n @JsName("floatClass")\n val floatClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(),
"Float", { it is Float })\n\n @JsName("doubleClass")\n val doubleClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(), "Double", { it is Double })\n\n @JsName("arrayClass")\n val arrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>>(), "Array", { it is Array<*> })\n\n @JsName("stringClass")\n val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(),
"String", { it is String })\n\n @JsName("throwableClass")\n val throwableClass =
PrimitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable })\n\n}

```

```

@JsName("booleanArrayClass")\n val booleanArrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<BooleanArray>>(), "BooleanArray", { it is BooleanArray
})\n\n @JsName("charArrayClass")\n val charArrayClass =
PrimitiveKClassImpl(js("Uint16Array").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray
})\n\n
@JsName("byteArrayClass")\n val byteArrayClass =
PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray })\n\n
@JsName("shortArrayClass")\n val shortArrayClass =
PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray
})\n\n @JsName("intArrayClass")\n val intArrayClass =
PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntArray>>(), "IntArray", { it is IntArray })\n\n
@JsName("longArrayClass")\n val longArrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray })\n\n
@JsName("floatArrayClass")\n val floatArrayClass =
PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray
})\n\n @JsName("doubleArrayClass")\n val doubleArrayClass =
PrimitiveKClassImpl(js("Float64Array").unsafeCast<JsClass<DoubleArray>>(), "DoubleArray",
{ it is DoubleArray })\n\n @JsName("functionClass")\n fun functionClass(arity: Int): KClassImpl<Any> {\n
return functionClasses.get(arity) ?: run {\n val result =
PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function$arity",\n
{ jsTypeOf(it) === "function" && it.asDynamic().length === arity })\n functionClasses.asDynamic()[arity]
= result\n result\n }\n }\n\nprivate val functionClasses =
arrayOfNulls<KClassImpl<Any>>(0), "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n a package is omitted to get declarations directly under the module\n\nimport
kotlin.reflect.*\nimport kotlin.reflect.js.internal.*\n\n@JsName("getKClass")\ninternal fun <T : Any>
getKClass(jClass: Any /* JsClass<T>
| Array<JsClass<T>> */): KClass<T> {\n return if (js("Array").isArray(jClass)) {\n
getKClassM(jClass.unsafeCast<Array<JsClass<T>>>())\n } else {\n
getKClass1(jClass.unsafeCast<JsClass<T>>())\n }\n}\n\n@JsName("getKClassM")\ninternal fun <T : Any>
getKClassM(jClasses: Array<JsClass<T>>): KClass<T> = when (jClasses.size) {\n 1 ->
getKClass1(jClasses[0])\n 0 -> NothingKClassImpl.unsafeCast<KClass<T>>()\n else ->
ErrorKClass().unsafeCast<KClass<T>>()\n}\n\n@JsName("getKClassFromExpression")\ninternal fun <T : Any>
getKClassFromExpression(e: T): KClass<T> =\n when (jsTypeOf(e)) {\n "string" ->
PrimitiveClasses.stringClass\n "number" -> if (jsBitwiseOr(e, 0).asDynamic() === e)
PrimitiveClasses.intClass else PrimitiveClasses.doubleClass\n "boolean" -> PrimitiveClasses.booleanClass\n
"function" -> PrimitiveClasses.functionClass(e.asDynamic().length)\n else -> {\n when {\n e
is BooleanArray
-> PrimitiveClasses.booleanArrayClass\n e is CharArray -> PrimitiveClasses.charArrayClass\n e
is ByteArray -> PrimitiveClasses.byteArrayClass\n e is ShortArray -> PrimitiveClasses.shortArrayClass\n
e is IntArray -> PrimitiveClasses.intArrayClass\n e is LongArray ->
PrimitiveClasses.longArrayClass\n e is FloatArray -> PrimitiveClasses.floatArrayClass\n e is
DoubleArray -> PrimitiveClasses.doubleArrayClass\n e is KClass<*> -> KClass::class\n e is
Array<*> -> PrimitiveClasses.arrayClass\n else -> {\n val constructor =
js("Object").getPrototypeOf(e).constructor\n when {\n constructor === js("Object") ->
PrimitiveClasses.anyClass\n constructor === js("Error") -> PrimitiveClasses.throwableClass\n
else -> {\n

```

```

 val jsClass: JsClass<T> = constructor\n getKClass1(jsClass)\n }\n }\n }\n }\n }.unsafeCast<KClass<T>>()\n\n@JsName("getKClass1")\n\ninternal fun <T : Any> getKClass1(jClass: JsClass<T>): KClass<T> {\n if (jClass === js("String")) return PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()\n\n val metadata = jClass.asDynamic().`$metadata$\`\n\n return if (metadata != null) {\n if (metadata.`$kClass$\` == null) {\n val kClass = SimpleKClassImpl(jClass)\n metadata.`$kClass$\` = kClass\n kClass\n } else {\n metadata.`$kClass$\`\n }\n } else {\n SimpleKClassImpl(jClass)\n }\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Exposes the JavaScript [RegExp] object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/RegExp) to Kotlin.\n\n */\n\n@Suppress("NOT_DOCUMENTED")\n\npublic external class RegExp(pattern: String, flags: String? = definedExternally) {\n\n public fun test(str: String): Boolean\n\n public fun exec(str: String): RegExpMatch?\n\n public override fun toString(): String\n\n /**\n * The lastIndex is a read/write integer property of regular expressions that specifies the index at which to start the next match.\n */\n\n public var lastIndex: Int\n\n public val global: Boolean\n\n public val ignoreCase: Boolean\n\n public val multiline: Boolean\n\n}\n\n/**\n * Resets the regular expression so that subsequent [RegExp.test] and [RegExp.exec] calls will match starting with the beginning of the input string.\n\n */\n\npublic fun RegExp.reset() {\n lastIndex = 0\n}\n\n// TODO: Inherit from array or introduce asArray() extension\n\n/**\n * Represents the return value of [RegExp.exec].\n\n */\n\n@Suppress("NOT_DOCUMENTED")\n\npublic external interface RegExpMatch {\n\n public val index: Int\n\n public val input: String\n\n public val length: Int\n\n}\n\n/**\n * Returns the entire text matched by [RegExp.exec] if the [index] parameter is 0, or the text matched by the capturing parenthesis\n * at the given index.\n\n */\n\npublic inline operator fun RegExpMatch.get(index: Int): String? = asDynamic()[index]\n\n/**\n * Converts the result of [RegExp.exec] to an array where the first element contains the entire matched text and each subsequent\n * element is the text matched by each capturing parenthesis.\n\n */\n\npublic inline fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String?>>()\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.sequences\n\n\ninternal actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {\n\n private var sequenceRef: Sequence<T>? = sequence\n\n actual override fun iterator(): Iterator<T> {\n\n val sequence = sequenceRef ?: throw IllegalStateException("This sequence can be consumed only once.")\n\n sequenceRef = null\n\n return sequence.iterator()\n }\n}\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n\n@SinceKotlin("1.5")\n\npublic actual enum class CharCategory(internal val value: Int, public actual val code: String) {\n\n /**\n * General category \"Cn\" in the Unicode specification.\n */\n\n UNASSIGNED(0, "Cn"),\n\n /**\n * General category \"Lu\" in the Unicode specification.\n */\n\n UPPERCASE_LETTER(1, "Lu"),\n\n /**\n * General category \"Ll\" in the Unicode specification.\n */\n\n LOWERCASE_LETTER(2, "Ll"),\n\n /**\n * General category \"Lt\" in the Unicode specification.\n */\n\n TITLECASE_LETTER(3, "Lt"),\n\n /**\n * General category \"Lm\" in the Unicode specification.\n */\n\n MODIFIER_LETTER(4, "Lm"),\n\n /**\n * General category \"Lo\" in the Unicode specification.\n */\n\n OTHER_LETTER(5, "Lo"),\n\n /**\n * General category \"Mn\" in the Unicode specification.\n */\n\n NON_SPACING_MARK(6, "Mn"),\n\n /**\n * General category \"Me\" in the Unicode specification.\n */\n\n ENCLOSING_MARK(7, "Me"),\n\n /**\n * General category \"Mc\" in the Unicode specification.\n */\n\n COMBINING_SPACING_MARK(8, "Mc"),\n\n /**\n * General category \"Nd\" in the Unicode specification.\n */\n\n DECIMAL_DIGIT_NUMBER(9, "Nd"),\n\n /**\n * General category \"Ni\" in the Unicode specification.\n */\n}

```

```

 *^n LETTER_NUMBER(10, \"NI\"),\n\n /**\n * General category \"No\" in the Unicode specification.\n
 *^n OTHER_NUMBER(11, \"No\"),\n\n /**\n * General category \"Zs\" in the Unicode specification.\n
 *^n SPACE_SEPARATOR(12, \"Zs\"),\n\n /**\n * General category \"Zl\" in the Unicode specification.\n
 *^n LINE_SEPARATOR(13, \"Zl\"),\n\n /**\n * General category \"Zp\" in the Unicode specification.\n
 *^n PARAGRAPH_SEPARATOR(14, \"Zp\"),\n\n /**\n * General category \"Cc\" in the Unicode
 specification.\n *^n CONTROL(15, \"Cc\"),\n\n /**\n * General category \"Cf\" in the Unicode
 specification.\n *^n FORMAT(16, \"Cf\"),\n\n /**\n * General category \"Co\" in the Unicode
 specification.\n *^n PRIVATE_USE(18, \"Co\"),\n\n /**\n * General category \"Cs\" in the Unicode
 specification.\n *^n SURROGATE(19, \"Cs\"),\n\n /**\n * General category \"Pd\" in
 the Unicode specification.\n *^n DASH_PUNCTUATION(20, \"Pd\"),\n\n /**\n * General category \"Ps\"
 in the Unicode specification.\n *^n START_PUNCTUATION(21, \"Ps\"),\n\n /**\n * General category
 \"Pe\" in the Unicode specification.\n *^n END_PUNCTUATION(22, \"Pe\"),\n\n /**\n * General category
 \"Pc\" in the Unicode specification.\n *^n CONNECTOR_PUNCTUATION(23, \"Pc\"),\n\n /**\n *
 General category \"Po\" in the Unicode specification.\n *^n OTHER_PUNCTUATION(24, \"Po\"),\n\n /**\n
 * General category \"Sm\" in the Unicode specification.\n *^n MATH_SYMBOL(25, \"Sm\"),\n\n /**\n *
 General category \"Sc\" in the Unicode specification.\n *^n CURRENCY_SYMBOL(26, \"Sc\"),\n\n /**\n
 * General category \"Sk\" in the Unicode specification.\n *^n MODIFIER_SYMBOL(27, \"Sk\"),\n\n /**\n
 * General category \"So\" in the Unicode specification.\n *^n OTHER_SYMBOL(28, \"So\"),\n\n
 /**\n * General category \"Pi\" in the Unicode specification.\n *^n
 INITIAL_QUOTE_PUNCTUATION(29, \"Pi\"),\n\n /**\n * General category \"Pf\" in the Unicode
 specification.\n *^n FINAL_QUOTE_PUNCTUATION(30, \"Pf\");\n\n /**\n * Returns `true` if [char]
 character belongs to this category.\n *^n public actual operator fun contains(char: Char): Boolean =
 char.getCategoryValue() == this.value\n\n companion object {\n internal fun valueOf(category: Int):
 CharCategory =\n when (category) {\n in 0..16 -> values()[category]\n in 18..30 ->
 values()[category - 1]\n else -> throw IllegalArgumentException(\"Category #\$category is not defined.\")\n
 }\n }\n\n\", \"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
 */\n\npackage
 kotlin.text\n\n/**\n * The exception thrown when a character encoding or decoding error occurs.\n
 */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual open class
 CharacterCodingException(message: String?) : Exception(message) {\n actual constructor() : this(null)\n}\n\n\", \"/*\n
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
 is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
 kotlin.text\n\n/**\n * A mutable sequence of characters.\n */\n * String builder can be used to efficiently perform
 multiple string manipulation operations.\n */\npublic actual class StringBuilder actual constructor(content: String) :
 Appendable, CharSequence {\n /**\n * Constructs an empty string builder with the specified initial [capacity].\n
 */\n * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the
 further performance of operations.\n */\n actual constructor(capacity: Int) : this() {\n }\n\n /**\n Constructs a
 string builder that contains the same characters as the specified [content] char sequence. */\n actual
 constructor(content: CharSequence) : this(content.toString()) {\n }\n\n /**\n Constructs an empty string builder. */\n
 actual constructor() : this(\"\")\n\n private var string: String = if (content !== undefined) content else \"\"\n\n
 actual override val length: Int\n get() = string.asDynamic().length\n\n actual override fun get(index: Int): Char
 =\n string.getOrElse(index) { throw IndexOutOfBoundsException(\"index: \$index, length: \$length\") }\n\n
 actual override fun subSequence(startIndex: Int, endIndex: Int): CharSequence = string.substring(startIndex,
 endIndex)\n\n actual override fun append(value: Char): StringBuilder {\n string += value\n return this\n
 }\n\n actual override fun append(value: CharSequence?):
 StringBuilder {\n string += value.toString()\n return this\n }\n\n actual override fun append(value:
 CharSequence?, startIndex: Int, endIndex: Int): StringBuilder =\n this.appendRange(value ?: \"null\", startIndex,

```



```

endIndex)\n\n /**\n * Reverses the contents of this string builder and returns this instance.\n *\n *
Surrogate pairs included in this string builder are treated as single characters.\n * Therefore, the order of the high-
low surrogates is never reversed.\n *\n * Note that the reverse operation may produce new surrogate pairs that
were unpaired low-surrogates and high-surrogates before the operation.\n * For example, reversing
`"\uDC00\uD800"` produces `"\uD800\uDC00"` which is a valid surrogate pair.\n *\n actual fun
reverse(): StringBuilder {\n var reversed = ""\n var index = string.length - 1\n while (index >= 0) {\n
 val low = string[index--]\n if (low.isLowSurrogate()
&& index >= 0) {\n val high = string[index--]\n if (high.isHighSurrogate()) {\n
reversed = reversed + high + low\n } else {\n reversed = reversed + low + high\n }\n
 } else {\n reversed += low\n }\n }\n string = reversed\n return this\n }\n\n
/**\n * Appends the string representation of the specified object [value] to this string builder and returns this
instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string by the
`value.toString()` method,\n * and then that string was appended to this string builder.\n *\n actual fun
append(value: Any?): StringBuilder {\n string += value.toString()\n return this\n }\n\n /**\n *
Appends the string representation of the specified boolean [value] to this string builder and returns this instance.\n
*\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()`
method,\n * and then that string was appended to this string builder.\n *\n @SinceKotlin("1.3")\n actual
fun append(value: Boolean): StringBuilder {\n string += value\n return this\n }\n\n /**\n * Appends
characters in the specified character array [value] to this string builder and returns this instance.\n *\n *
Characters are appended in order, starting at the index 0.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun append(value: CharArray): StringBuilder {\n
string += value.concatToString()\n return this\n }\n\n @Deprecated("Provided for binary compatibility.",
level = DeprecationLevel.HIDDEN)\n fun append(value: String): StringBuilder = append(value)\n\n /**\n *
Appends the specified string [value] to this string builder and returns
this instance.\n *\n * If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are appended.\n *\n
@SinceKotlin("1.3")\n actual fun append(value: String?): StringBuilder {\n this.string += value ?: "\u0000\u0000\u0000\u0000"\n
return this\n }\n\n /**\n * Returns the current capacity of this string builder.\n *\n * The capacity is the
maximum length this string builder can have before an allocation occurs.\n *\n * In Kotlin/JS implementation
of StringBuilder the value returned from this method may not indicate the actual size of the backing storage.\n
*\n @SinceKotlin("1.3")\n // @ExperimentalStdlibApi\n @Deprecated("Obtaining StringBuilder capacity is
not supported in JS and common code.", level = DeprecationLevel.ERROR)\n actual fun capacity(): Int =
length\n\n /**\n * Ensures that the capacity of this string builder is at least equal to the specified
[minimumCapacity].\n *\n * If the current capacity
is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n * Otherwise, this
method takes no action and simply returns.\n *\n * In Kotlin/JS implementation of StringBuilder the size of the
backing storage is not extended to comply the given [minimumCapacity],\n * thus calling this method has no
effect on the further performance of operations.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun ensureCapacity(minimumCapacity: Int) {\n
}\n\n /**\n * Returns the index within this string builder of the first occurrence of the specified [string].\n *\n
* Returns -1 if the specified [string] does not occur in this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun indexOf(string: String): Int =
this.string.asDynamic().indexOf(string)\n\n /**\n * Returns the index within this string builder of
the first occurrence of the specified [string],\n * starting at the specified [startIndex].\n *\n * Returns -1 if
the specified [string] does not occur in this string builder starting at the specified [startIndex].\n *\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun indexOf(string: String,
startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)\n\n /**\n * Returns the index within
this string builder of the last occurrence of the specified [string].\n * The last occurrence of empty string `""` is
considered to be at the index equal to `this.length`.\n *\n * Returns -1 if the specified [string] does not occur

```

```

in this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun lastIndexOf(string: String): Int = this.string.asDynamic().lastIndexOf(string)\n /**\n * Returns the index within this string builder of the last occurrence of the specified [string],\n * starting from the specified [startIndex] toward the beginning.\n *\n * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun lastIndexOf(string: String, startIndex: Int): Int {\n if (string.isEmpty() && startIndex < 0) return -1\n return this.string.asDynamic().lastIndexOf(string, startIndex)\n }\n /**\n * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: Boolean): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0, index) + value + string.substring(index)\n return this\n }\n /**\n * Inserts the specified character [value] into this string builder at the specified [index] and returns this instance.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: Char): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0, index) + value + string.substring(index)\n return this\n }\n /**\n * Inserts characters in the specified character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in same order as in the [value] character array, starting at [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: CharArray): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0, index) + value.concatToString() + string.substring(index)\n return this\n }\n /**\n * Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which characters are inserted. If [value] is `null`, then the four characters `\\null` are inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: CharSequence?): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0, index) + value.toString() + string.substring(index)\n return this\n }\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: Any?): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0, index) + value.toString() + string.substring(index)\n return this\n }\n @Deprecated("Provided for binary compatibility.", level = DeprecationLevel.HIDDEN)\n fun insert(index: Int, value: String): StringBuilder = insert(index, value)\n /**\n * Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n *\n * If [value] is `null`, then the four characters `\\null` are inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length

```

```

of this string builder.\n *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: String?): String {
 val toInsert = value ?: "\u0000"
 this.string = this.string.substring(0, index) + toInsert +
 this.string.substring(index)\n return this\n }\n /**\n * Sets the length of this string builder to the
 specified [newLength].\n * If the [newLength] is less than the current length, it is changed to the specified
 [newLength].\n * Otherwise, null characters '\u0000' are appended to this string builder until its length is less
 than the [newLength].\n * Note that in Kotlin/JS [set] operator function has non-constant execution time
 complexity.\n * Therefore, increasing length of this string builder and then updating each character by index may
 slow down your program.\n * @throws
 IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n *^n
 @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun setLength(newLength:
 Int) {\n if (newLength < 0) {\n throw IllegalArgumentException("Negative new length:
 $newLength.")\n }\n if (newLength <= length) {\n string = string.substring(0, newLength)\n }
 else {\n for (i in length until newLength) {\n string += "\u0000"\n }\n }\n }\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the
 [length] (exclusive).\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater
 than the length of this string builder.\n *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun substring(startIndex: Int):
 String {\n AbstractList.checkPositionIndex(startIndex, length)\n return string.substring(startIndex)\n }\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and
 up to the [endIndex] (exclusive).\n * @throws IndexOutOfBoundsException or
 [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when
 `startIndex > endIndex`.\n *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun substring(startIndex: Int, endIndex: Int): String {\n AbstractList.checkBoundsIndexes(startIndex,
 endIndex, length)\n return string.substring(startIndex, endIndex)\n }\n /**\n * Attempts to reduce
 storage used for this string builder.\n * If the backing storage of this string builder is larger than necessary
 to hold its current contents,\n * then it may be resized to become more
 space efficient.\n * Calling this method may, but is not required to, affect the value of the [capacity] property.\n
 *^n * In Kotlin/JS implementation of String Builder the size of the backing storage is always equal to the length of
 the string builder.\n *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun trimToSize() {\n }\n override fun toString(): String = string\n /**\n * Clears the content of
 this string builder making it empty and returns this instance.\n * @sample
 samples.text.Strings.clearStringBuilder\n *^n @SinceKotlin("1.3")\n public fun clear(): String {
 }\n /**\n * Sets the character at the specified [index] to the specified
 [value].\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n
 *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public operator fun set(index: Int, value: Char) {\n AbstractList.checkElementIndex(index, length)\n string = string.substring(0, index) + value + string.substring(index + 1)\n }\n /**\n * Replaces characters in
 the specified range of this string builder with characters in the specified string [value] and returns this instance.\n
 *^n * @param startIndex the beginning (inclusive) of the range to replace.\n * @param endIndex the end
 (exclusive) of the range to replace.\n * @param value the string to replace with.\n * @throws
 IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length
 of this string builder, or `startIndex > endIndex`.\n *^n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun setRange(startIndex: Int, endIndex: Int, value:
 String): String {
 checkReplaceRange(startIndex, endIndex, length)\n this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n return this\n }\n private fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n if (startIndex < 0 ||
 startIndex > length) {\n throw IndexOutOfBoundsException("startIndex: $startIndex, length: $length")\n }\n }\n }

```

```

}\n if (startIndex > endIndex) {\n throw IllegalArgumentException("\n startIndex($startIndex) >
endIndex($endIndex)")\n }\n }\n\n /**\n * Removes the character at the specified [index] from this string
builder and returns this instance.\n *\n * If the `Char` at the specified [index] is part of a supplementary code
point, this method does not remove the entire supplementary character.\n *\n * @param index the index of
`Char` to remove.\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string
builder.\n *\n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteAt(index: Int): StringBuilder {\n
AbstractList.checkElementIndex(index, length)\n\n string = string.substring(0, index) + string.substring(index +
1)\n return this\n }\n\n /**\n * Removes characters in the specified range from this string builder and
returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to remove.\n *
@param endIndex the end (exclusive) of the range to remove.\n *\n * @throws IndexOutOfBoundsException
or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex >
endIndex`.\n *\n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n public fun
deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n checkReplaceRange(startIndex, endIndex,
length)\n\n string = string.substring(0, startIndex) + string.substring(endIndex)\n
return this\n }\n\n /**\n * Copies characters from this string builder into the [destination] character
array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the
array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by
default.\n * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is
out of range of this string builder indices or when `startIndex > endIndex`.\n * @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified
[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n *\n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n public
fun toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length)
{\n AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n
AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size)\n\n
var dstIndex = destinationOffset\n for (index in startIndex until endIndex) {\n destination[dstIndex++]
= string[index]\n }\n }\n\n /**\n * Appends characters in a subarray of the specified character array
[value] to this string builder and returns this instance.\n *\n * Characters are appended in order, starting at
specified [startIndex].\n *\n * @param value the array from which characters are appended.\n * @param
startIndex the beginning (inclusive) of the subarray to append.\n * @param endIndex the end (exclusive) of the
subarray to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException]
when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n
*\n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n public fun
appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n string +=
value.concatToString(startIndex, endIndex)\n return this\n }\n\n /**\n * Appends a subsequence of the
specified character sequence [value] to this string builder and returns this instance.\n *\n * @param value the
character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of
the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of the [value] character sequence indices or when `startIndex > endIndex`.\n *\n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun
appendRange(value: CharSequence, startIndex:
Int, endIndex: Int): StringBuilder {\n val stringCsq = value.toString()\n
AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n\n string +=
stringCsq.substring(startIndex, endIndex)\n return this\n }\n\n /**\n * Inserts characters in a subarray of
the specified character array [value] into this string builder at the specified [index] and returns this instance.\n
*\n * The inserted characters go in same order as in the [value] array, starting at [index].\n *\n * @param index

```

the position in this string builder to insert at.\n \* @param value the array from which characters are inserted.\n \* @param startIndex the beginning (inclusive) of the subarray to insert.\n \* @param endIndex the end (exclusive) of the subarray to insert.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n \* @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n AbstractList.checkPositionIndex(index, this.length)\n string = string.substring(0, index) + value.concatToString(startIndex, endIndex) + string.substring(index)\n return this\n }\n /\*\*\n \* Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n \* The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n \* @param index the position in this string builder to insert at.\n \* @param value the character sequence from which a subsequence is inserted.\n \* @param startIndex the beginning (inclusive) of the subsequence to insert.\n \* @param endIndex the end (exclusive) of the subsequence to insert.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n \* @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n val stringCsq = value.toString()\n AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n string = string.substring(0, index) + stringCsq.substring(startIndex, endIndex) + string.substring(index)\n return this\n }\n /\*\*\n \* Clears the content of this string builder making it empty and returns this instance.\n \* @sample samples.text.Strings.clearStringBuilder\n \* @SinceKotlin("1.3")\n @Suppress("EXTENSION\_SHADOWED\_BY\_MEMBER", "NOTHING\_TO\_INLINE")\n public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()\n /\*\*\n \* Sets the character at the specified [index] to the specified [value].\n \* @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n \* @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("EXTENSION\_SHADOWED\_BY\_MEMBER", "NOTHING\_TO\_INLINE")\n public actual inline operator fun StringBuilder.set(index: Int, value: Char) = this.set(index, value)\n /\*\*\n \* Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n \* @param startIndex the beginning (inclusive) of the range to replace.\n \* @param endIndex the end (exclusive) of the range to replace.\n \* @param value the string to replace with.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n \* @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("EXTENSION\_SHADOWED\_BY\_MEMBER", "NOTHING\_TO\_INLINE")\n public actual inline fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder =\n this.setRange(startIndex, endIndex, value)\n /\*\*\n \* Removes the character at the specified [index] from this string builder and returns this instance.\n \* If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n \* @param index the index of `Char` to remove.\n \* @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n \* @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("EXTENSION\_SHADOWED\_BY\_MEMBER", "NOTHING\_TO\_INLINE")\n public actual inline fun StringBuilder.deleteAt(index: Int): StringBuilder = this.deleteAt(index)\n /\*\*\n \* Removes characters in the specified range from this string builder and returns this instance.\n \* @param startIndex the beginning (inclusive) of the range to remove.\n \* @param endIndex the end (exclusive) of the range to remove.

@param endIndex the end (exclusive) of the range to remove.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder
= this.deleteRange(startIndex, endIndex)\n\n**\n * Copies characters from this string builder into the [destination]
character array.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the
array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n
* @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n *\n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of this string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException
when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when
that index is out of the [destination] array indices range.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE",
"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun
StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
this.length) =\n this.toCharArray(destination, destinationOffset, startIndex, endIndex)\n\n**\n * Appends
characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n *
Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which
characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param
endIndex the end (exclusive) of the subarray to append.\n *\n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER",
"NOTHING_TO_INLINE")\npublic actual inline fun StringBuilder.appendRange(value: CharArray, startIndex:
Int, endIndex: Int): StringBuilder =\n this.appendRange(value, startIndex, endIndex)\n\n**\n * Appends a
subsequence of the specified character sequence [value] to this string builder and returns this instance.\n *\n *
@param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning
(inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out
of range of the [value] character sequence indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex:
Int): StringBuilder =\n this.appendRange(value, startIndex, endIndex)\n\n**\n * Inserts characters in a subarray
of the specified character array [value] into this string builder at the specified [index] and returns this instance.\n
*\n * The inserted characters go in same order as in the [value] array, starting at [index].\n *\n * @param index the
position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n *
@param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive) of
the subarray to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER",

```

```

\`NOTHING_TO_INLINE\`)npublic actual inline fun StringBuilder.insertRange(index: Int, value: CharArray,
startIndex: Int, endIndex: Int): StringBuilder =\n this.insertRange(index, value, startIndex, endIndex)\n\n/*\n * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified
[index] and returns this instance.\n * \n * The inserted characters go in the same order as in the [value] character
sequence, starting at [index].\n * \n * @param index the position in this string builder to insert at.\n * @param value
the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the
subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex >
endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this
string builder.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n
this.insertRange(index, value, startIndex, endIndex)\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n * \n\npackage kotlin.text\n\n/**\n * Returns `true` if the content of this
string is equal to the word `true`, ignoring case, and `false` otherwise.\n * \n\n@Deprecated("Use Kotlin compiler
1.4 to avoid deprecation warning.")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@kotlin.internal.InlineOnly\npublic actual
inline fun String.toBoolean(): Boolean = this.toBoolean()\n\n/**\n * Returns `true` if this string is not `null` and its
content is equal to the word `true`, ignoring case, and `false` otherwise.\n * \n * There are also strict versions of the
function available on non-nullable String, [toBooleanStrict] and [toBooleanStrictOrNull].\n
*/\n\n@SinceKotlin("1.4")npublic actual fun String?.toBoolean(): Boolean = this != null && this.lowercase() ==
`true`\n\n/**\n * Parses the string as a signed [Byte] number and returns the result.\n * @throws
NumberFormatException if the string is not a valid representation of a number.\n\npublic actual fun
String.toByte(): Byte = toByteOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a signed [Byte]
number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a
number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*/\n\npublic
actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the
string as a [Short] number and returns the result.\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n\npublic actual fun String.toShort(): Short = toShortOrNull() ?:
numberFormatError(this)\n\n/**\n * Parses the string as a [Short] number and returns the result.\n * @throws
NumberFormatException if the string is not a valid representation of a number.\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\npublic actual fun
String.toShort(radix: Int): Short = toShortOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as
an [Int] number and returns the result.\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n\npublic actual fun String.toInt(): Int = toIntOrNull() ?:
numberFormatError(this)\n\n/**\n * Parses the
string as an [Int] number and returns the result.\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to
number conversion.\n\npublic actual fun String.toInt(radix: Int): Int = toIntOrNull(radix) ?:
numberFormatError(this)\n\n/**\n * Parses the string as a [Long] number and returns the result.\n * @throws
NumberFormatException if the string is not a valid representation of a number.\n\npublic actual fun
String.toLong(): Long = toLongOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [Long]
number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a
number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n

```

```

*^/npublic actual fun String.toLong(radix: Int): Long = toLongOrNull(radix) ?: numberFormatException(this)\n\n/**\n *
Parses the string
as a [Double] number and returns the result.\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n */npublic actual fun String.toDouble(): Double =
+(this.asDynamic()).unsafeCast<Double>().also {\n if (it.isNaN() && !this.isNaN() || it == 0.0 &&
this.isBlank())\n numberFormatException(this)\n}\n\n/**\n * Parses the string as a [Float] number and returns the
result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*/n@kotlin.internal.InlineOnly\npublic actual inline fun String.toFloat(): Float =
toDouble().unsafeCast<Float>()\n\n/**\n * Parses the string as a [Double] number and returns the result\n * or `null`
if the string is not a valid representation of a number.\n */npublic actual fun String.toDoubleOrNull(): Double? =
+(this.asDynamic()).unsafeCast<Double>().takeIf {\n !(it.isNaN() && !this.isNaN() || it == 0.0 &&
this.isBlank())\n}\n\n/**\n * Parses the string as a [Float] number
and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*/n@kotlin.internal.InlineOnly\npublic actual inline fun String.toFloatOrNull(): Float? =
toDoubleOrNull().unsafeCast<Float?>()\n\n/**\n * Returns a string representation of this [Byte] value in the
specified [radix].\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string
conversion.\n */n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Byte.toString(radix:
Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string representation of this [Short] value in the
specified [radix].\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string
conversion.\n */n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Short.toString(radix:
Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string representation of this [Int] value in the specified
[radix].\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string
conversion.\n */n@SinceKotlin("1.2")\npublic actual fun Int.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))\n\nprivate fun String.isNaN(): Boolean = when (this.lowercase()) {\n
"nan", "+nan", "-nan" -> true\n else -> false\n}\n\n/**\n * Checks whether the given [radix] is valid radix for
string to number and number to string conversion.\n */n@PublishedApi\ninternal actual fun checkRadix(radix: Int):
Int {\n if (radix !in 2..36) {\n throw IllegalArgumentException("\radix $radix was not in valid range 2..36")\n
}\n return radix\n}\n\ninternal actual fun digitOf(char: Char, radix: Int): Int = when {\n char >= '0' && char <=
'9' -> char - '0'\n char >= 'A' && char <= 'Z' -> char - 'A' + 10\n char >= 'a' && char <= 'z' -> char - 'a' + 10\n
char < "\u0080" -> -1\n char >= "\uFF21" && char <= "\uFF3A" -> char - "\uFF21"
+ 10 // full-width latin capital letter\n char >= "\uFF41" && char <= "\uFF5A" -> char - "\uFF41" + 10 // full-width
latin small letter\n else -> char.digitToIntImpl()\n}.let { if (it >= radix) -1 else it }\n", "/"\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */n\npackage kotlin.text\n\nimport
kotlin.js.RegExp\n\n/**\n * Provides enumeration values to use to set regular expression options.\n */npublic actual
enum class RegexOptions(val value: String) {\n /** Enables case-insensitive matching. */n
IGNORE_CASE("i"),\n /** Enables multiline mode.\n * In multiline mode the expressions `^` and `$`
match just after or just before,\n * respectively, a line terminator or the end of the input sequence. */n
MULTILINE("m")\n}\n\nprivate fun Iterable<RegexOption>.toFlags(prepend: String): String
= joinToString("\",", prefix = prepend) { it.value }\n\n/**\n * Represents the results from a single capturing group
within a [MatchResult] of [Regex].\n * @param value The value of captured group.\n */npublic actual data
class MatchGroup(actual val value: String)\n\n/**\n * Represents a compiled regular expression.\n * Provides
functions to match strings in text with a pattern, replace the found occurrences and split text around matches.\n *
For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions)\n
*/n * and
[http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).\n
*/n * Note that `RegExp` objects under the hood are constructed with [the `u`

```



flag](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/RegExp/unicode)\n \* that enables Unicode-related features in regular

expressions. This also makes the pattern syntax more strict,\n \* for example, prohibiting unnecessary escape sequences.\n \* \n \* @constructor Creates a regular expression from the specified [pattern] string and the specified set of [options].\n \* \n public actual class Regex actual constructor(pattern: String, options: Set<RegexOption>) {\n \* \n /\*\* Creates a regular expression from the specified [pattern] string and the specified single [option].\n \* \n public actual constructor(pattern: String, option: RegexOption) : this(pattern, setOf(option))\n \* \n /\*\* Creates a regular expression from the specified [pattern] string and the default options.\n \* \n public actual constructor(pattern: String) : this(pattern, emptySet())\n \* \n /\*\* The pattern string of this regular expression.\n \* \n public actual val pattern: String = pattern\n \* \n /\*\* The set of options that were used to create this regular expression.\n \* \n public actual val options: Set<RegexOption> = options.toSet()\n

```
private val nativePattern: RegExp = RegExp(pattern, options.toFlags("gu"))\n private var nativeStickyPattern: RegExp? = null\n private fun initStickyPattern(): RegExp =\n nativeStickyPattern ?: RegExp(pattern, options.toFlags("yu")).also { nativeStickyPattern = it }\n private var nativeMatchesEntirePattern: RegExp? = null\n private fun initMatchesEntirePattern(): RegExp =\n nativeMatchesEntirePattern ?: run {\n if (pattern.startsWith('^') && pattern.endsWith('$'))\n nativePattern\n else\n return\n RegExp("^${pattern.trimStart('^').trimEnd('$')}$", options.toFlags("gu"))\n }.also {\n nativeMatchesEntirePattern = it\n }\n /** Indicates whether the regular expression matches the entire [input].\n * \n public actual infix fun matches(input: CharSequence): Boolean {\n nativePattern.reset()\n val match = nativePattern.exec(input.toString())\n return match != null && match.index == 0 && nativePattern.lastIndex == input.length\n }\n /** Indicates whether the regular expression can find at least one match in the specified [input].\n * \n public actual fun containsMatchIn(input: CharSequence): Boolean {\n nativePattern.reset()\n return nativePattern.test(input.toString())\n }\n @SinceKotlin("1.5")\n @ExperimentalStdlibApi\n public actual fun matchesAt(input: CharSequence, index: Int): Boolean {\n if (index < 0 || index > input.length) {\n throw IndexOutOfBoundsException("index out of bounds: $index, input length: ${input.length}")\n }\n val pattern = initStickyPattern()\n pattern.lastIndex = index\n return pattern.test(input.toString())\n }\n /**\n * Returns the first match of a regular expression in the [input], beginning at the specified [startIndex].\n * \n * @param startIndex An index to start search with, by default 0. Must be not less than zero and not greater than `input.length`\n * \n * @return An instance of [MatchResult] if match was found or `null` otherwise.\n * \n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of the [input] char sequence.\n * \n * @sample samples.text.Regexp.find\n * \n @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun find(input: CharSequence, startIndex: Int = 0): MatchResult? {\n if (startIndex < 0 || startIndex > input.length) {\n throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")\n }\n return nativePattern.findNext(input.toString(), startIndex, nativePattern)\n }\n /**\n * Returns a sequence of all occurrences of a regular expression within the [input] string, beginning at the specified [startIndex].\n * \n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of the [input] char sequence.\n * \n * @sample samples.text.Regexp.findAll\n * \n @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun findAll(input: CharSequence, startIndex: Int = 0): Sequence<MatchResult> {\n if (startIndex < 0 || startIndex > input.length) {\n throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")\n }\n return generateSequence({ find(input, startIndex) }, { match -> match.next() })\n }\n /**\n * Attempts to match the entire [input] CharSequence against the pattern.\n * \n * @return An instance of [MatchResult] if the entire input matches or `null` otherwise.\n * \n public actual fun matchEntire(input: CharSequence): MatchResult? =\n initMatchesEntirePattern().findNext(input.toString(), 0, nativePattern)\n @SinceKotlin("1.5")\n @ExperimentalStdlibApi\n public actual fun matchAt(input:
```

CharSequence, index: Int):

```
MatchResult? {\n if (index < 0 || index > input.length) {\n throw IndexOutOfBoundsException("index\n out of bounds: $index, input length: ${input.length}")\n }\n return\n initStickyPattern().findNext(input.toString(), index, nativePattern)\n }\n\n /**\n * Replaces all occurrences\n of this regular expression in the specified [input] string with specified [replacement] expression.\n *\n * The\n replacement string may contain references to the captured groups during a match. Occurrences of `index`\n *\n * in\n the replacement string will be substituted with the subsequences corresponding to the captured groups with the\n specified index.\n *\n * The first digit after '$' is always treated as part of group reference. Subsequent digits are\n incorporated\n *\n * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered\n as potential components\n *\n * of the group reference. Note that indexes of captured\n groups start from 1, and the group with index 0 is the whole match.\n *\n * Backslash character '\\' can be used\n to include the succeeding character as a literal in the replacement string, e.g, '\\$` or '\\\\\\'.\n *\n * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n *\n * Note that\n referring named capturing groups by name is currently not supported in Kotlin/JS.\n *\n * However, you can still\n refer them by index.\n *\n * @param input the char sequence to find matches of this regular expression in\n *\n * @param replacement the expression to replace found matches with\n *\n * @return the result of replacing each\n occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression\n *\n * @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or\n `index` does not exist\n */\n public actual fun replace(input: CharSequence,\n replacement: String): String {\n if (!replacement.contains("\\\\\\\\") && !replacement.contains('$')) {\n return\n input.toString().nativeReplace(nativePattern, replacement)\n }\n return replace(input) {\n substituteGroupRefs(it, replacement) }\n }\n\n /**\n * Replaces all occurrences of this regular expression in\n the specified [input] string with the result of\n *\n * the given function [transform] that takes [MatchResult] and\n returns a string to be used as a\n *\n * replacement for that match.\n */\n public actual fun replace(input:\n CharSequence, transform: (MatchResult) -> CharSequence): String {\n var match = find(input)\n if (match\n == null) return input.toString()\n var lastStart = 0\n val length = input.length\n val sb =\n StringBuilder(length)\n do {\n val foundMatch = match!!\n sb.append(input, lastStart,\n foundMatch.range.start)\n sb.append(transform(foundMatch))\n lastStart = foundMatch.range.endInclusive + 1\n match = foundMatch.next()\n } while (lastStart\n < length && match != null)\n if (lastStart < length) {\n sb.append(input, lastStart, length)\n }\n return sb.toString()\n }\n\n /**\n * Replaces the first occurrence of this regular expression in the specified\n [input] string with specified [replacement] expression.\n *\n * The replacement string may contain references to\n the captured groups during a match. Occurrences of `index`\n *\n * in the replacement string will be substituted with\n the subsequences corresponding to the captured groups with the specified index.\n *\n * The first digit after '$' is\n always treated as part of group reference. Subsequent digits are incorporated\n *\n * into `index` only if they would\n form a valid group reference. Only the digits '0'..'9' are considered as potential components\n *\n * of the group\n reference. Note that indexes\n of captured groups start from 1, and the group with index 0 is the whole match.\n *\n * Backslash character '\\' can be used\n to include the succeeding character as a literal in the replacement string, e.g, '\\$` or '\\\\\\'.\n *\n * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n *\n * Note that\n referring named capturing groups by name is not supported currently in Kotlin/JS.\n *\n * However, you can still\n refer them by index.\n *\n * @param input the char sequence to find a match of this regular expression in\n *\n * @param replacement the expression to replace the found match with\n *\n * @return the result of replacing the first\n occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression\n *\n * @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or\n `index` does not exist\n */\n public actual fun\n replaceFirst(input: CharSequence, replacement: String): String {\n if (!replacement.contains("\\\\\\\\") &&\n !replacement.contains('$')) {\n val nonGlobalOptions = options.toFlags("u")\n return
```





pairs that were unpaired low-surrogates and high-surrogates before the operation. \n \* For example, reversing `\"uDC00\"uD800\"` produces `\"uD800\"uDC00\"` which is a valid surrogate pair. \n \*/ \n fun reverse(): StringBuilder \n \n /\*\* \n \* Appends the string representation of the specified object [value] to this string builder and returns this instance. \n \* \n \* The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method, \n \* and then that string was appended to this string builder. \n \* \n fun append(value: Any?): StringBuilder \n \n /\*\* \n \* Appends the string representation of the specified boolean [value] to this string builder and returns this instance. \n \* \n \* The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method, \n \* and then that string was appended to this string builder. \n \* \n @SinceKotlin(\"1.3\") \n fun append(value: Boolean): StringBuilder \n \n /\*\* \n \* Appends characters in the specified character array [value] to this string builder and returns this instance. \n \* \n \* Characters are appended in order, starting at the index 0. \n \* \n @SinceKotlin(\"1.4\") \n @WasExperimental(ExperimentalStdlibApi::class) \n fun append(value: CharArray): StringBuilder \n \n /\*\* \n \* Appends the specified string [value] to this string builder and returns this instance. \n \* \n \* If [value] is `null`, then the four characters `\"null\"` are appended. \n \* \n @SinceKotlin(\"1.3\") \n fun append(value: String?): StringBuilder \n \n /\*\* \n \* Returns the current capacity of this string builder. \n \* \n \* The capacity is the maximum length this string builder can have before an allocation occurs. \n \* \n @SinceKotlin(\"1.3\") \n // @ExperimentalStdlibApi \n @Deprecated(\"Obtaining StringBuilder capacity is not supported in JS and common code.\", level = DeprecationLevel.ERROR) \n fun capacity(): Int \n \n /\*\* \n \* Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity]. \n \* \n \* If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity. \n \* \n \* Otherwise, this method takes no action and simply returns. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun ensureCapacity(minimumCapacity: Int) \n \n /\*\* \n \* Returns the index within this string builder of the first occurrence of the specified [string]. \n \* \n \* Returns `-1` if the specified [string] does not occur in this string builder. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun indexOf(string: String): Int \n \n /\*\* \n \* Returns the index within this string builder of the first occurrence of the specified [string], \n \* starting at the specified [startIndex]. \n \* \n \* Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex]. \n \* \n @SinceKotlin(\"1.4\") \n @WasExperimental(ExperimentalStdlibApi::class) \n fun indexOf(string: String, startIndex: Int): Int \n \n /\*\* \n \* Returns the index within this string builder of the last occurrence of the specified [string]. \n \* \n \* The last occurrence of empty string `\"\"` is considered to be at the index equal to `this.length`. \n \* \n \* Returns `-1` if the specified [string] does not occur in this string builder. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun lastIndexOf(string: String): Int \n \n /\*\* \n \* Returns the index within this string builder of the last occurrence of the specified [string], \n \* starting from the specified [startIndex] toward the beginning. \n \* \n \* Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex]. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun lastIndexOf(string: String, startIndex: Int): Int \n \n /\*\* \n \* Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance. \n \* \n \* The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method, \n \* and then that string was inserted into this string builder at the specified [index]. \n \* \n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun insert(index: Int, value: Boolean): StringBuilder \n \n /\*\* \n \* Inserts the specified character [value] into this string builder at the specified [index] and returns this instance. \n \* \n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder. \n \* \n @SinceKotlin(\"1.4\") \n \n @WasExperimental(ExperimentalStdlibApi::class) \n fun insert(index: Int, value: Char): StringBuilder \n \n /\*\* \n \* Inserts characters in the specified character array

```

[value] into this string builder at the specified [index] and returns this
instance.\n * \n * The inserted characters go in same order as in the [value] character array, starting at
[index].\n * \n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of
this string builder.\n * \n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun
insert(index: Int, value: CharArray): StringBuilder\n\n /**\n * Inserts characters in the specified character
sequence [value] into this string builder at the specified [index] and returns this instance.\n * \n * The inserted
characters go in the same order as in the [value] character sequence, starting at [index].\n * \n * @param index
the position in this string builder to insert at.\n * @param value the character sequence from which characters are
inserted. If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n * \n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater
than the length of this string builder.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharSequence?):
StringBuilder\n\n /**\n * Inserts the string representation of the specified object [value] into this string builder
at the specified [index] and returns this instance.\n * \n * The overall effect is exactly as if the [value] were
converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder
at the specified [index].\n * \n * @throws IndexOutOfBoundsException if [index] is less than zero or greater
than the length of this string builder.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Any?): StringBuilder\n\n /**\n
* Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n * \n * If
[value]
is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n * \n * @throws IndexOutOfBoundsException if
[index] is less than zero or greater than the length of this string builder.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: String?): StringBuilder\n\n
/**\n * Sets the length of this string builder to the specified [newLength].\n * \n * If the [newLength] is less
than the current length, it is changed to the specified [newLength].\n * Otherwise, null characters `'\u0000'` are
appended to this string builder until its length is less than the [newLength].\n * \n * Note that in Kotlin/JS [set]
operator function has non-constant execution time complexity.\n * Therefore, increasing length of this string
builder and then updating each character by index may slow down your program.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] if [newLength]
is less than zero.\n * \n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun
setLength(newLength: Int)\n\n /**\n * Returns a new [String] that contains characters in this string builder at
[startIndex] (inclusive) and up to the [length] (exclusive).\n * \n * @throws IndexOutOfBoundsException if
[startIndex] is less than zero or greater than the length of this string builder.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun substring(startIndex: Int): String\n\n /**\n * Returns
a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [endIndex]
(exclusive).\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n
fun substring(startIndex: Int, endIndex: Int): String\n\n /**\n * Attempts to reduce storage used for this string
builder.\n * \n * If the backing storage of this string builder is larger than necessary to hold its current
contents,\n * then it may be resized to become more space efficient.\n * Calling this method may, but is not
required to, affect the value of the [capacity] property.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun trimToSize()\n}\n\n\n/**\n * Clears the content of this
string builder making it empty and returns this instance.\n * \n * @sample samples.text.Strings.clearStringBuilder\n
\n\n@SinceKotlin("1.3")\npublic expect fun StringBuilder.clear(): String\n\n\n/**\n * Sets the character at
the specified [index] to the specified [value].\n * \n * @throws IndexOutOfBoundsException if [index] is out of
bounds of this string builder.\n
\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic

```

expect operator fun `StringBuilder.set(index: Int, value: Char)` Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance. @param startIndex the beginning (inclusive) of the range to replace. @param endIndex the end (exclusive) of the range to replace. @param value the string to replace with. @throws `IndexOutOfBoundsException` or `IllegalArgumentException` if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder` Removes the character at the specified [index] from this string builder and returns this instance. If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character. @param index the index of `Char` to remove. @throws `IndexOutOfBoundsException` if [index] is out of bounds of this string builder.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.deleteAt(index: Int): StringBuilder` Removes characters in the specified range from this string builder and returns this instance. @param startIndex the beginning (inclusive) of the range to remove. @param endIndex the end (exclusive) of the range to remove. @throws `IndexOutOfBoundsException` or `IllegalArgumentException` when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder` Copies characters from this string builder into the [destination] character array. @param destination the array to copy to. @param destinationOffset the position in the array to copy to, 0 by default. @param startIndex the beginning (inclusive) of the range to copy, 0 by default. @param endIndex the end (exclusive) of the range to copy, length of this string builder by default. @throws `IndexOutOfBoundsException` or `IllegalArgumentException` when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`. @throws `IndexOutOfBoundsException` when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length)` Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance. Characters are appended in order, starting at specified [startIndex]. @param value the array from which characters are appended. @param startIndex the beginning (inclusive) of the subarray to append. @param endIndex the end (exclusive) of the subarray to append. @throws `IndexOutOfBoundsException` or `IllegalArgumentException` when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder` Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance. @param value the character sequence from which a subsequence is appended. @param startIndex the beginning (inclusive) of the subsequence to append. @param endIndex the end (exclusive) of the subsequence to append. @throws `IndexOutOfBoundsException` or `IllegalArgumentException` when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun `StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder` Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance. The inserted characters go in same order as in the [value] array, starting at [index]. @param index the position in this string builder to insert at. @param value the array from which characters

are inserted.\n \* @param startIndex the beginning (inclusive) of the subarray to insert.\n \* @param endIndex the end (exclusive) of the subarray to insert.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n \* \n \* @SinceKotlin("1.4")\n \* @WasExperimental(ExperimentalStdlibApi::class)\n \* public expect fun StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n \* \n \* Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n \* \n \* The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n \* \n \* @param index the position in this string builder to insert at.\n \* @param value the character sequence from which a subsequence is inserted.\n \* @param startIndex the beginning (inclusive) of the subsequence to insert.\n \* @param endIndex the end (exclusive) of the subsequence to insert.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n \* \n \* @SinceKotlin("1.4")\n \* @WasExperimental(ExperimentalStdlibApi::class)\n \* public expect fun StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder\n \* \n \* @Suppress("EXTENSION\_SHADOWED\_BY\_MEMBER")\n \* @Deprecated("Use append(value: Any?) instead", ReplaceWith("append(value = obj)"), DeprecationLevel.WARNING)\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.append(obj: Any?): StringBuilder = this.append(obj)\n \* \n \* Builds new string by populating newly created [StringBuilder] using provided [builderAction]\n \* \n \* and then converting it to [String].\n \* \n \* @kotlin.internal.InlineOnly\n \* public inline fun buildString(builderAction: StringBuilder.() -> Unit): String {\n \* contract { callsInPlace(builderAction, InvocationKind.EXACTLY\_ONCE) }\n \* return StringBuilder().apply(builderAction).toString()\n \* }\n \* \n \* Builds new string by populating newly created [StringBuilder] initialized with the given [capacity]\n \* \n \* using provided [builderAction] and then converting it to [String].\n \* \n \* @SinceKotlin("1.1")\n \* @kotlin.internal.InlineOnly\n \* public inline fun buildString(capacity: Int, builderAction: StringBuilder.() -> Unit): String {\n \* contract { callsInPlace(builderAction, InvocationKind.EXACTLY\_ONCE) }\n \* return StringBuilder(capacity).apply(builderAction).toString()\n \* }\n \* \n \* Appends all arguments to the given StringBuilder.\n \* \n \* @public fun StringBuilder.append(vararg value: String?): StringBuilder {\n \* for (item in value)\n \* append(item)\n \* return this\n \* }\n \* \n \* Appends all arguments to the given StringBuilder.\n \* \n \* @public fun StringBuilder.append(vararg value: Any?): StringBuilder {\n \* for (item in value)\n \* append(item)\n \* return this\n \* }\n \* \n \* Appends a line feed character (`\n`) to this StringBuilder. \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(): StringBuilder = append("\n")\n \* \n \* Appends [value] to this [StringBuilder], followed by a line feed character (`\n`). \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(value: CharSequence?): StringBuilder = append(value).appendLine()\n \* \n \* Appends [value] to this [StringBuilder], followed by a line feed character (`\n`). \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(value: String?): StringBuilder = append(value).appendLine()\n \* \n \* Appends [value] to this [StringBuilder], followed by a line feed character (`\n`). \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(value: Any?): StringBuilder = append(value).appendLine()\n \* \n \* Appends [value] to this [StringBuilder], followed by a line feed character (`\n`). \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(value: CharArray): StringBuilder = append(value).appendLine()\n \* \n \* Appends [value] to this [StringBuilder], followed by a line feed character (`\n`). \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline fun StringBuilder.appendLine(value: Char): StringBuilder = append(value).appendLine()\n \* \n \* Appends [value] to this



[StringBuilder], followed by a line feed character (`\n`).

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value:
Boolean): StringBuilder = append(value).appendLine()\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source
code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.text\nimport kotlin.js.RegExp\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun
String.nativeIndexOf(ch: Char, fromIndex: Int): Int = nativeIndexOf(ch.toString(),
fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeLastIndexOf(ch: Char,
fromIndex: Int): Int = nativeLastIndexOf(ch.toString(), fromIndex)\n\n/**\n * Returns `true` if this string starts with
the specified prefix.\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return
nativeStartsWith(prefix, 0)\n else\n return regionMatches(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n if
(!ignoreCase)\n return nativeStartsWith(prefix, startIndex)\n else\n return regionMatches(startIndex,
prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this string ends with the specified suffix.\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return
nativeEndsWith(suffix)\n else\n return regionMatches(length - suffix.length, suffix, 0, suffix.length,
ignoreCase)\n}\n\n@Deprecated("Use Regex.matches() instead",
ReplaceWith("regex.toRegex().matches(this)"))\n@DeprecatedSinceKotlin(warningSince = "1.6")\npublic fun
String.matches(regex: String): Boolean {\n @Suppress("DEPRECATION")\n val result = this.match(regex)\n return result != null
&& result.size != 0\n}\n\n/**\n * Returns `true` if this string is empty or consists solely of whitespace characters.\n */\n@sample samples.text.Strings.stringIsBlank\n\n@public actual fun CharSequence.isBlank(): Boolean = length
== 0 || indices.all { this[it].isWhitespace() }\n\n/**\n * Returns `true` if this string is equal to [other], optionally
ignoring character case.\n */\n * Two strings are considered to be equal if they have the same length and the same
character at the same index.\n */\n * If [ignoreCase] is true, the result of `Char.uppercaseChar().lowercaseChar()` on
each character is compared.\n */\n * @param ignoreCase `true` to ignore character case when comparing strings. By
default `false`.\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual
fun String?.equals(other: String?, ignoreCase: Boolean = false): Boolean {\n if (this == null) return other == null\n if (other == null) return false\n if (!ignoreCase) return this == other\n if (this.length
!= other.length) return false\n for (index in 0 until this.length) {\n val thisChar = this[index]\n val
otherChar = other[index]\n if (!thisChar.equals(otherChar, ignoreCase)) {\n return false\n }\n }\n return true\n}\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual
fun CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase:
Boolean = false): Boolean =\n regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)\n\n/**\n * Returns a copy of this string having its first letter titlecased using the rules of the default locale,\n */\n * or the original
string if it's empty or already starts with a title case letter.\n */\n * The title case of a character is usually the same as
its upper case with several exceptions.\n */\n * The particular list of characters with the special title case form depends
on the underlying platform.\n */\n * @sample samples.text.Strings.capitalize\n\n@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase())
it.titlecase() else it.toString() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic actual fun
String.capitalize(): String {\n return if (isEmpty()) substring(0, 1).uppercase() + substring(1) else
this\n}\n\n/**\n * Returns a copy of this string having its first letter lowercased using the rules of the default
locale,\n */\n * or the original string if it's empty or already starts with a lower case letter.\n */\n * @sample
samples.text.Strings.decapitalize\n */\n@Deprecated("Use replaceFirstChar instead.",
```

```

ReplaceWith("replaceFirstChar { it.lowercase() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
actual fun String.decapitalize(): String {\n return if (isEmpty()) substring(0, 1).lowercase() + substring(1) else
this\n}\n\n/**\n * Returns a string containing this char sequence repeated [n] times.\n * @throws
[IllegalArgumentException]
 when n < 0.\n * @sample samples.text.Strings.repeat\n */\npublic actual fun CharSequence.repeat(n: Int): String
{\n require(n >= 0) {\n "Count 'n' must be non-negative, but was $n." }\n return when (n) {\n 0 -> ""\n 1 -> this.toString()\n else -> {\n var result = ""\n if (!isEmpty()) {\n var s =
this.toString()\n var count = n\n while (true) {\n if ((count and 1) == 1) {\n
 result += s\n }\n count = count ushr 1\n if (count == 0) {\n
 break\n }\n s += s\n }\n }\n return result\n }\n }\n}\n\n/**\n * Returns a new string obtained by replacing all occurrences of the [oldValue] substring in this string\n * with the
specified [newValue] string.\n * @sample samples.text.Strings.replace\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) "gui" else "gu"),
 Regex.nativeEscapeReplacement(newValue))\n\n/**\n * Returns a new string with all occurrences of [oldChar]
replaced with [newChar].\n * @sample samples.text.Strings.replace\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) "gui" else "gu"),
 newChar.toString())\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual
fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) "ui" else "u"),
 Regex.nativeEscapeReplacement(newValue))\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGU
MENTS")\npublic
 actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) "ui" else "u"),
 newChar.toString())\n\n"/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/** Returns the negative [size] if [throwOnMalformed] is
false, throws [CharacterCodingException] otherwise. */\nprivate fun malformed(size: Int, index: Int,
throwOnMalformed: Boolean): Int {\n if (throwOnMalformed) throw CharacterCodingException("Malformed
sequence starting at ${index - 1}")\n return -size\n}\n\n/** Returns code point corresponding to UTF-16
surrogate pair,\n * where the first of the pair is the [high] and the second
 is in the [string] at the [index].\n * Returns zero if the pair is malformed and [throwOnMalformed] is false.\n * @throws CharacterCodingException if the pair is malformed and [throwOnMalformed] is true.\n */\nprivate fun
codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n if (high !in 0xD800..0xDBFF || index >= endIndex) {\n return malformed(0, index, throwOnMalformed)\n }\n val low = string[index].code\n if (low !in 0xDC00..0xDFFF) {\n return malformed(0, index,
throwOnMalformed)\n }\n return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of two bytes,\n * where the first byte of the sequence is the
[byte1] and the second byte is in the [bytes] array at the [index].\n * Returns zero if the sequence is malformed and
[throwOnMalformed] is false.\n * @throws CharacterCodingException if the sequence of two bytes is
malformed and [throwOnMalformed] is true.\n */\nprivate fun codePointFrom2(bytes: ByteArray, byte1: Int, index:
Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n if (byte1 and 0x1E == 0 || index >= endIndex) {\n
 return malformed(0, index, throwOnMalformed)\n }\n val byte2 = bytes[index].toInt()\n if (byte2 and 0xC0 !=
0x80) {\n return malformed(0, index, throwOnMalformed)\n }\n return (byte1 shl 6) xor byte2 xor
0xF80\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of three bytes,\n * where the first byte of

```

the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n \* Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence\n \* if the sequence is malformed and [throwOnMalformed] is false.\n \* @throws CharacterCodingException if the sequence of three bytes is malformed and [throwOnMalformed] is true.\n \*/\nprivate fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n if (index >= endIndex) {\n return malformed(0, index, throwOnMalformed)\n }\n val byte2 = bytes[index].toInt()\n if (byte1 and 0xF == 0) {\n if (byte2 and 0xE0 != 0xA0) {\n // Non-shortest form\n return malformed(0, index, throwOnMalformed)\n } else if (byte1 and 0xF == 0xD) {\n if (byte2 and 0xE0 != 0x80) {\n // Surrogate code point\n return malformed(0, index, throwOnMalformed)\n } else if (byte2 and 0xC0 != 0x80) {\n return malformed(0, index, throwOnMalformed)\n }\n }\n if (index + 1 == endIndex) {\n return malformed(1, index, throwOnMalformed)\n }\n val byte3 = bytes[index + 1].toInt()\n if (byte3 and 0xC0 != 0x80) {\n return malformed(1, index, throwOnMalformed)\n }\n return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -0x1E080\n }\n}\n\n \* Returns code point corresponding to UTF-8 sequence of four bytes,\n \* where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n \* Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence\n \* if the sequence is malformed and [throwOnMalformed] is false.\n \* @throws CharacterCodingException if the sequence of four bytes is malformed and [throwOnMalformed] is true.\n \*/\nprivate fun codePointFrom4(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n if (index >= endIndex) {\n return malformed(0, index, throwOnMalformed)\n }\n val byte2 = bytes[index].toInt()\n if (byte1 and 0xF == 0x0) {\n if (byte2 and 0xF0 <= 0x80) {\n // Non-shortest form\n return malformed(0, index, throwOnMalformed)\n } else if (byte1 and 0xF == 0x4) {\n if (byte2 and 0xF0 != 0x80) {\n // Out of Unicode code points domain (larger than U+10FFFF)\n return malformed(0, index, throwOnMalformed)\n } else if (byte1 and 0xF > 0x4) {\n return malformed(0, index, throwOnMalformed)\n } else if (byte2 and 0xC0 != 0x80) {\n return malformed(0, index, throwOnMalformed)\n }\n }\n if (index + 1 == endIndex) {\n return malformed(1, index, throwOnMalformed)\n }\n val byte3 = bytes[index + 1].toInt()\n if (byte3 and 0xC0 != 0x80) {\n return malformed(1, index, throwOnMalformed)\n }\n if (index + 2 == endIndex) {\n return malformed(2, index, throwOnMalformed)\n }\n val byte4 = bytes[index + 2].toInt()\n if (byte4 and 0xC0 != 0x80) {\n return malformed(2, index, throwOnMalformed)\n }\n return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80\n }\n}\n\n \* Maximum number of bytes needed to encode a single char.\n \* Code points in `0..0x7F` are encoded in a single byte.\n \* Code points in `0x80..0x7FF` are encoded in two bytes.\n \* Code points in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.\n \* Surrogate code points in `0xD800..0xDFFF` are not Unicode scalar values, therefore aren't encoded.\n \* Code points in `0x10000..0x10FFFF` are represented by a pair of surrogate `Char`s and are encoded in four bytes.\n \*/\nprivate const val MAX\_BYTES\_PER\_CHAR = 3\n\n \* The byte sequence a malformed UTF-16 char sequence is replaced by.\n \*/\nprivate val REPLACEMENT\_BYTE\_SEQUENCE: ByteArray = byteArrayOf(0xEF.toByte(), 0xBF.toByte(), 0xBD.toByte())\n\n \* Encodes the [string] using UTF-8 and returns the resulting [ByteArray].\n \* @param string the string to encode.\n \* @param startIndex the start offset (inclusive) of the substring to encode.\n \* @param endIndex the end offset (exclusive) of the substring to encode.\n \* @param throwOnMalformed whether to throw on malformed char sequence or replace by the [REPLACEMENT\_BYTE\_SEQUENCE].\n \*/\n\n \* @throws CharacterCodingException if the char sequence is malformed and [throwOnMalformed] is true.\n \*/\ninternal fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): ByteArray {\n require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)\n val bytes = ByteArray((endIndex - startIndex) \* MAX\_BYTES\_PER\_CHAR)\n var byteIndex = 0\n var charIndex = startIndex\n while (charIndex < endIndex) {\n val code = string[charIndex++].code\n when {\n code < 0x80 -> bytes[byteIndex++] = code.toByte()\n code < 0x800 -> {\n bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()\n bytes[byteIndex++] = ((code and 0x3F) or



```

were\n * suppressed in order to deliver this exception.\n *\n@SinceKotlin("1.4")\npublic actual fun
Throwable.addSuppressed(exception:
Throwable) {\n if (this !== exception) {\n val suppressed =
this.asDynamic()._suppressed.unsafeCast<MutableList<Throwable>?>()\n if (suppressed == null) {\n
this.asDynamic()._suppressed = mutableListOf(exception)\n } else {\n suppressed.add(exception)\n
}\n }\n}\n\n/**\n * Returns a list of all exceptions that were suppressed in order to deliver this exception.\n
*\n@SinceKotlin("1.4")\npublic actual val Throwable.suppressedExceptions: List<Throwable>\n get() {\n
return this.asDynamic()._suppressed?.unsafeCast<List<Throwable>>() ?: emptyList()\n }\n\nprivate class
ExceptionTraceBuilder {\n private val target = StringBuilder()\n private val visited = arrayOf<Throwable>()\n
private var topStack: String = ""\n private var topStackStart: Int = 0\n fun buildFor(exception: Throwable):
String {\n exception.dumpFullTrace("", "")\n return target.toString()\n }\n\n private fun
Throwable.dumpFullTrace(indent: String, qualifier: String) {\n this.dumpSelfTrace(indent, qualifier) ||
return\n\n var cause = this.cause\n while (cause != null) {\n cause.dumpSelfTrace(indent, "Caused
by: ") || return\n cause = cause.cause\n }\n }\n\n private fun Throwable.dumpSelfTrace(indent:
String, qualifier: String): Boolean {\n target.append(indent).append(qualifier)\n val shortInfo =
this.toString()\n if (hasSeen(this)) {\n target.append("[CIRCULAR REFERENCE, SEE ABOVE:
").append(shortInfo).append("\\n")\n return false\n }\n visited.asDynamic().push(this)\n var
stack = this.asDynamic().stack as String?\n if (stack != null) {\n val stackStart =
stack.indexOf(shortInfo).let { if (it < 0) 0 else it + shortInfo.length }\n
if (stackStart == 0) target.append(shortInfo).append("\\n")\n if (topStack.isEmpty()) {\n
topStack = stack\n topStackStart = stackStart\n } else {\n stack =
dropCommonFrames(stack, stackStart)\n }\n if (indent.isNotEmpty()) {\n // indent stack, but
avoid indenting exception message lines\n val messageLines = if (stackStart == 0) 0 else 1 +
shortInfo.count { c -> c == '\\n' }\n stack.lineSequence().forEachIndexed { index: Int, line: String ->}\n
if (index >= messageLines) target.append(indent)\n target.append(line).append("\\n")\n
}\n } else {\n target.append(stack).append("\\n")\n }\n } else {\n
target.append(shortInfo).append("\\n")\n }\n\n val suppressed = suppressedExceptions\n if
(suppressed.isNotEmpty()) {\n
val suppressedIndent = indent + " \n" for (s in suppressed) {\n
s.dumpFullTrace(suppressedIndent, "Suppressed: ")\n }\n }\n return true\n }\n\n private fun
dropCommonFrames(stack: String, stackStart: Int): String {\n var commonFrames: Int = 0\n var lastBreak:
Int = 0\n var preLastBreak: Int = 0\n for (pos in 0 until minOf(topStack.length - topStackStart, stack.length
- stackStart)) {\n val c = stack[stack.lastIndex - pos]\n if (c != topStack[topStack.lastIndex - pos])
break\n if (c == '\\n') {\n commonFrames += 1\n preLastBreak = lastBreak\n
lastBreak = pos\n }\n }\n if (commonFrames <= 1) return stack\n while (preLastBreak > 0 &&
stack[stack.lastIndex - (preLastBreak - 1)] == ' '\n preLastBreak -= 1\n // leave 1 common frame to
ease matching with the top exception
stack\n return stack.dropLast(preLastBreak) + "... and ${commonFrames - 1} more common stack frames
skipped"\n }\n }"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
\n@npackage kotlin.time\nimport kotlin.js.json\nimport kotlin.math.\n\ninternal actual inline val
durationAssertionsEnabled: Boolean get() = true\n\ninternal actual fun formatToExactDecimals(value: Double,
decimals: Int): String {\n val rounded = if (decimals == 0) {\n value\n } else {\n val pow =
10.0.pow(decimals)\n JsMath.round(abs(value) * pow) / pow * sign(value)\n }\n return if (abs(rounded) <
1e21) {\n // toFixed switches to scientific format after 1e21\n
rounded.asDynamic().toFixed(decimals).unsafeCast<String>()\n } else {\n // toPrecision outputs the specified
number of digits,

```

```

but only for positive numbers\n val positive = abs(rounded)\n val positiveString =
positive.asDynamic().toPrecision(ceil(log10(positive)) + decimals).unsafeCast<String>()\n if (rounded < 0) \"-
$positiveString\" else positiveString\n }\n}\n\ninternal actual fun formatUpToDecimals(value: Double, decimals:
Int): String {\n return value.asDynamic().toLocaleString(\"en-us\", json(\"maximumFractionDigits\" to
decimals)).unsafeCast<String>()\n}\n\n\"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.time\n\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n\npublic actual enum class
DurationUnit(internal val scale: Double) {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of
a microsecond.\n */\n NANOSECONDS(1e0),\n /**\n * Time unit representing
one microsecond, which is 1/1000 of a millisecond.\n */\n MICROSECONDS(1e3),\n /**\n * Time unit
representing one millisecond, which is 1/1000 of a second.\n */\n MILLISECONDS(1e6),\n /**\n * Time
unit representing one second.\n */\n SECONDS(1e9),\n /**\n * Time unit representing one minute.\n */\n
 MINUTES(60e9),\n /**\n * Time unit representing one hour.\n */\n HOURS(3600e9),\n /**\n * Time
unit representing one day, which is always equal to 24 hours.\n */\n DAYS(86400e9);\n}\n\n@SinceKotlin(\"1.3\")\n\ninternal actual fun convertDurationUnit(value: Double, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Double {\n val sourceCompareTarget =
sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> value *
(sourceUnit.scale / targetUnit.scale)\n sourceCompareTarget < 0 -> value / (targetUnit.scale /
sourceUnit.scale)\n }\n}\n\n@SinceKotlin(\"1.5\")\n\ninternal actual fun convertDurationUnitOverflow(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Long {\n val sourceCompareTarget =
sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> value *
(sourceUnit.scale / targetUnit.scale).toLong()\n sourceCompareTarget < 0 -> value / (targetUnit.scale /
sourceUnit.scale).toLong()\n }\n}\n\n@SinceKotlin(\"1.5\")\n\ninternal actual fun
convertDurationUnit(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n val
sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> {\n val scale = (sourceUnit.scale / targetUnit.scale).toLong()\n val result
= value * scale\n when {\n result / scale == value -> result\n value > 0 ->
Long.MAX_VALUE\n }\n }\n sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n }\n}\n}\n\n\"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport org.w3c.performance.GlobalPerformance\nimport
org.w3c.performance.Performance\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n\ninternal actual object
MonotonicTimeSource : TimeSource {\n private val actualSource: TimeSource = run {\n val isNode:
Boolean = js(\"typeof process !== 'undefined' && process.versions && !!process.versions.node\")\n if
(isNode)\n HrTimeSource(js(\"process\").unsafeCast<Process>())\n else\n js(\"self\").unsafeCast<GlobalPerformance?>().performance?.let(::PerformanceTimeSource)\n }\n\n DateNowTimeSource\n}\n\n override fun markNow(): TimeMark = actualSource.markNow()\n}\n\n\ninternal external interface Process
{\n fun hrtime(time: Array<Double> = definedExternally):
Array<Double>\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n\ninternal class HrTimeSource(val process:
Process) : TimeSource {\n override fun markNow(): TimeMark = object : TimeMark() {\n val startedAt =
process.hrtime()\n override fun elapsedNow(): Duration =\n process.hrtime(startedAt).let { (seconds,
nanos) -> seconds.toDuration(DurationUnit.SECONDS) + nanos.toDuration(DurationUnit.NANOSECONDS) }\n }\n\n override fun toString(): String =
\"TimeSource(process.hrtime())\"\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n\ninternal class

```



```

get() = NodeType == Node.ELEMENT_NODE\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/** Removes all
the children from this node. */\n@SinceKotlin("1.4")\npublic
fun Node.clear() {\n while (hasChildNodes()) {\n removeChild(firstChild!!)\n }\n}\n\n/**\n * Creates text
node and append it to the element.\n */\n * @return this element\n */\n@SinceKotlin("1.4")\nfun
Element.appendText(text: String): Element {\n appendChild(ownerDocument!!.createTextNode(text))\n return
this\n}\n\n", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage org.w3c.dom\n\n@Deprecated("Use UnionMessagePortOrWindowProxy instead.")\n
ReplaceWith("UnionMessagePortOrWindowProxy")\n\ntypealias UnionMessagePortOrWindow =
UnionMessagePortOrWindowProxy\n\n@Deprecated("Use `as` instead.", ReplaceWith("`as`"))\n\nvar
HTMLLinkElement.as_\n get() = `as`\n set(value) {\n `as` = value\n }\n\n@Deprecated("Use `is`
instead.", ReplaceWith("`is`"))\n\nvar
ElementCreationOptions.is_\n get() = `is`\n set(value) {\n `is` = value\n }\n\n", /*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.khronos.webgl\n\nimport kotlin.js.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external
interface WebGLContextAttributes {\n var alpha: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var depth: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var stencil: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var antialias: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var premultipliedAlpha: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var preserveDrawingBuffer: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var preferLowPowerToHighPerformance: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var failIfMajorPerformanceCaveat:
Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun WebGLContextAttributes(alpha:
Boolean? = true, depth: Boolean? = true, stencil: Boolean? = false, antialias: Boolean? = true, premultipliedAlpha:
Boolean? = true, preserveDrawingBuffer: Boolean? = false, preferLowPowerToHighPerformance: Boolean? = false,
failIfMajorPerformanceCaveat: Boolean? = false): WebGLContextAttributes {\n
 val o = js("{}")\n o["alpha"] = alpha\n o["depth"] = depth\n o["stencil"] = stencil\n o["antialias"] =
antialias\n o["premultipliedAlpha"] = premultipliedAlpha\n o["preserveDrawingBuffer"] =
preserveDrawingBuffer\n o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance\n
 o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat\n return o\n}\n\npublic external abstract
class WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin\n */\npublic external
abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin\n */\npublic
external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin\n */\npublic external
abstract class WebGLProgram
: WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin\n */\npublic
external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin\n */\npublic external

```



```

abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin\n */\npublic external
abstract class WebGLTexture : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin\n
*/\npublic external abstract class WebGLUniformLocation\n\n/**\n * Exposes the JavaScript
[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin\n */\npublic
external abstract class WebGLActiveInfo {\n open val size: Int\n open val type: Int\n
 open val name: String\n}\n\n/**\n * Exposes the JavaScript
[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to
Kotlin\n */\npublic external abstract class WebGLShaderPrecisionFormat {\n open val rangeMin: Int\n open val
rangeMax: Int\n open val precision:
Int\n}\n\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface
WebGLRenderingContextBase {\n val canvas: HTMLCanvasElement\n val drawingBufferWidth: Int\n val
drawingBufferHeight: Int\n fun getContextAttributes(): WebGLContextAttributes?\n fun isContextLost():
Boolean\n fun getSupportedExtensions(): Array<String>?\n fun getExtension(name: String): dynamic\n fun
activeTexture(texture: Int)\n fun attachShader(program: WebGLProgram?, shader: WebGLShader?)\n fun
bindAttribLocation(program: WebGLProgram?, index: Int, name: String)\n fun bindBuffer(target: Int, buffer:
WebGLBuffer?)\n fun bindFramebuffer(target: Int,
framebuffer: WebGLFramebuffer?)\n fun bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)\n
fun bindTexture(target: Int, texture: WebGLTexture?)\n fun blendColor(red: Float, green: Float, blue: Float, alpha:
Float)\n fun blendEquation(mode: Int)\n fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)\n fun
blendFunc(sfactor: Int, dfactor: Int)\n fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha:
Int)\n fun bufferData(target: Int, size: Int, usage: Int)\n fun bufferData(target: Int, data: BufferDataSource?,
usage: Int)\n fun bufferSubData(target: Int, offset: Int, data: BufferDataSource?)\n fun
checkFramebufferStatus(target: Int): Int\n fun clear(mask: Int)\n fun clearColor(red: Float, green: Float, blue:
Float, alpha: Float)\n fun clearDepth(depth: Float)\n fun clearStencil(s: Int)\n fun colorMask(red: Boolean,
green: Boolean, blue: Boolean, alpha: Boolean)\n fun compileShader(shader: WebGLShader?)\n
 fun compressedTexImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, data:
ArrayBufferView)\n fun compressedTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int,
height: Int, format: Int, data: ArrayBufferView)\n fun copyTexImage2D(target: Int, level: Int, internalformat: Int,
x: Int, y: Int, width: Int, height: Int, border: Int)\n fun copyTexSubImage2D(target: Int, level: Int, xoffset: Int,
yoffset: Int, x: Int, y: Int, width: Int, height: Int)\n fun createBuffer(): WebGLBuffer?\n fun createFramebuffer():
WebGLFramebuffer?\n fun createProgram(): WebGLProgram?\n fun createRenderbuffer():
WebGLRenderbuffer?\n fun createShader(type: Int): WebGLShader?\n fun createTexture(): WebGLTexture?\n
fun cullFace(mode: Int)\n fun deleteBuffer(buffer: WebGLBuffer?)\n fun deleteFramebuffer(framebuffer:
WebGLFramebuffer?)\n fun deleteProgram(program: WebGLProgram?)\n fun deleteRenderbuffer(renderbuffer:
WebGLRenderbuffer?)\n fun deleteShader(shader: WebGLShader?)\n fun deleteTexture(texture:
WebGLTexture?)\n fun depthFunc(func: Int)\n fun depthMask(flag: Boolean)\n fun depthRange(zNear: Float,
zFar: Float)\n fun detachShader(program: WebGLProgram?, shader: WebGLShader?)\n fun disable(cap: Int)\n
fun disableVertexArray(index: Int)\n fun drawArrays(mode: Int, first: Int, count: Int)\n fun
drawElements(mode: Int, count: Int, type: Int, offset: Int)\n fun enable(cap: Int)\n fun
enableVertexArray(index: Int)\n fun finish()\n fun flush()\n fun framebufferRenderbuffer(target: Int,
attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)\n fun framebufferTexture2D(target:
Int, attachment: Int, textarget: Int, texture: WebGLTexture?, level: Int)\n fun frontFace(mode: Int)\n fun
generateMipmap(target: Int)\n fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n
fun getActiveUniform(program:
WebGLProgram?, index: Int): WebGLActiveInfo?\n fun getAttachedShaders(program: WebGLProgram?):
Array<WebGLShader>?\n fun getAttribLocation(program: WebGLProgram?, name: String): Int\n fun

```

```

getBufferParameter(target: Int, pname: Int): Any?\n fun getParameter(pname: Int): Any?\n fun getError(): Int\n
fun getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Any?\n fun
getProgramParameter(program: WebGLProgram?, pname: Int): Any?\n fun getProgramInfoLog(program:
WebGLProgram?): String?\n fun getRenderbufferParameter(target: Int, pname: Int): Any?\n fun
getShaderParameter(shader: WebGLShader?, pname: Int): Any?\n fun getShaderPrecisionFormat(shaderType: Int,
precisionType: Int): WebGLShaderPrecisionFormat?\n fun getShaderInfoLog(shader: WebGLShader?): String?\n
fun getShaderSource(shader: WebGLShader?): String?\n fun getTexParameter(target: Int, pname: Int): Any?\n
fun getUniform(program: WebGLProgram?,
location: WebGLUniformLocation?): Any?\n fun getUniformLocation(program: WebGLProgram?, name:
String): WebGLUniformLocation?\n fun glVertexAttrib(index: Int, pname: Int): Any?\n fun
getVertexAttribOffset(index: Int, pname: Int): Int\n fun hint(target: Int, mode: Int)\n fun isBuffer(buffer:
WebGLBuffer?): Boolean\n fun isEnabled(cap: Int): Boolean\n fun isFramebuffer(framebuffer:
WebGLFramebuffer?): Boolean\n fun isProgram(program: WebGLProgram?): Boolean\n fun
isRenderbuffer(renderbuffer: WebGLRenderbuffer?): Boolean\n fun isShader(shader: WebGLShader?): Boolean\n
fun isTexture(texture: WebGLTexture?): Boolean\n fun lineWidth(width: Float)\n fun linkProgram(program:
WebGLProgram?)\n fun pixelStorei(pname: Int, param: Int)\n fun polygonOffset(factor: Float, units: Float)\n
fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun
renderbufferStorage(target: Int, internalformat: Int,
width: Int, height: Int)\n fun sampleCoverage(value: Float, invert: Boolean)\n fun scissor(x: Int, y: Int, width:
Int, height: Int)\n fun shaderSource(shader: WebGLShader?, source: String)\n fun stencilFunc(func: Int, ref: Int,
mask: Int)\n fun stencilFuncSeparate(face: Int, func: Int, ref: Int, mask: Int)\n fun stencilMask(mask: Int)\n fun
stencilMaskSeparate(face: Int, mask: Int)\n fun stencilOp(fail: Int, zfail: Int, zpass: Int)\n fun
stencilOpSeparate(face: Int, fail: Int, zfail: Int, zpass: Int)\n fun texImage2D(target: Int, level: Int, internalformat:
Int, width: Int, height: Int, border: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun texImage2D(target:
Int, level: Int, internalformat: Int, format: Int, type: Int, source: TexImageSource?)\n fun texParameterf(target: Int,
pname: Int, param: Float)\n fun texParameteri(target: Int, pname: Int, param: Int)\n fun texSubImage2D(target:
Int, level: Int, xoffset: Int, yoffset: Int,
width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun texSubImage2D(target: Int, level:
Int, xoffset: Int, yoffset: Int, format: Int, type: Int, source: TexImageSource?)\n fun uniform1f(location:
WebGLUniformLocation?, x: Float)\n fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)\n
fun uniform1fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform1i(location:
WebGLUniformLocation?, x: Int)\n fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)\n fun
uniform1iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun uniform2f(location:
WebGLUniformLocation?, x: Float, y: Float)\n fun uniform2fv(location: WebGLUniformLocation?, v:
Float32Array)\n fun uniform2fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun
uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)\n fun uniform2iv(location:
WebGLUniformLocation?, v: Int32Array)\n fun uniform2iv(location: WebGLUniformLocation?,
v: Array<Int>)\n fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)\n fun
uniform3fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform3fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z:
Int)\n fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform3iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float, w: Float)\n fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)\n fun
uniform4fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform4i(location:
WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)\n fun uniform4iv(location: WebGLUniformLocation?, v:
Int32Array)\n fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n

```

```

fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
useProgram(program: WebGLProgram?)\n fun validateProgram(program: WebGLProgram?)\n fun
vertexAttrib1f(index: Int, x: Float)\n fun vertexAttrib1fv(index: Int, values: dynamic)\n fun
vertexAttrib2f(index: Int, x: Float, y: Float)\n fun vertexAttrib2fv(index: Int, values: dynamic)\n fun
vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)\n fun vertexAttrib3fv(index: Int, values: dynamic)\n fun
vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)\n
fun vertexAttrib4fv(index: Int, values: dynamic)\n fun vertexAttribPointer(index: Int, size: Int, type: Int,
normalized: Boolean, stride: Int, offset: Int)\n fun viewport(x: Int, y: Int, width: Int, height: Int)\n\n companion
object {\n val DEPTH_BUFFER_BIT: Int\n val STENCIL_BUFFER_BIT: Int\n val
COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES: Int\n val LINE_LOOP: Int\n val
LINE_STRIP: Int\n val TRIANGLES: Int\n val TRIANGLE_STRIP: Int\n val TRIANGLE_FAN:
Int\n val ZERO: Int\n val ONE: Int\n val SRC_COLOR: Int\n val ONE_MINUS_SRC_COLOR:
Int\n val SRC_ALPHA: Int\n val ONE_MINUS_SRC_ALPHA: Int\n val DST_ALPHA: Int\n val
ONE_MINUS_DST_ALPHA: Int\n val DST_COLOR: Int\n val ONE_MINUS_DST_COLOR: Int\n
val SRC_ALPHA_SATURATE: Int\n val FUNC_ADD: Int\n val BLEND_EQUATION: Int\n val
BLEND_EQUATION_RGB: Int\n
 val BLEND_EQUATION_ALPHA: Int\n val FUNC_SUBTRACT: Int\n val
FUNC_REVERSE_SUBTRACT: Int\n val BLEND_DST_RGB: Int\n val BLEND_SRC_RGB: Int\n
val BLEND_DST_ALPHA: Int\n val BLEND_SRC_ALPHA: Int\n val CONSTANT_COLOR: Int\n
val ONE_MINUS_CONSTANT_COLOR: Int\n val CONSTANT_ALPHA: Int\n val
ONE_MINUS_CONSTANT_ALPHA: Int\n val BLEND_COLOR: Int\n val ARRAY_BUFFER: Int\n
val ELEMENT_ARRAY_BUFFER: Int\n val ARRAY_BUFFER_BINDING: Int\n val
ELEMENT_ARRAY_BUFFER_BINDING: Int\n val STREAM_DRAW: Int\n val STATIC_DRAW: Int\n
val DYNAMIC_DRAW: Int\n val BUFFER_SIZE: Int\n val BUFFER_USAGE: Int\n val
CURRENT_VERTEX_ATTRIB: Int\n val FRONT: Int\n val BACK: Int\n val FRONT_AND_BACK:
Int\n val CULL_FACE: Int\n val BLEND: Int\n val DITHER: Int\n val STENCIL_TEST: Int\n
val DEPTH_TEST: Int\n val
SCISSOR_TEST: Int\n val POLYGON_OFFSET_FILL: Int\n val SAMPLE_ALPHA_TO_COVERAGE:
Int\n val SAMPLE_COVERAGE: Int\n val NO_ERROR: Int\n val INVALID_ENUM: Int\n val
INVALID_VALUE: Int\n val INVALID_OPERATION: Int\n val OUT_OF_MEMORY: Int\n val CW:
Int\n val CCW: Int\n val LINE_WIDTH: Int\n val ALIASED_POINT_SIZE_RANGE: Int\n val
ALIASED_LINE_WIDTH_RANGE: Int\n val CULL_FACE_MODE: Int\n val FRONT_FACE: Int\n
val DEPTH_RANGE: Int\n val DEPTH_WRITEMASK: Int\n val DEPTH_CLEAR_VALUE: Int\n val
DEPTH_FUNC: Int\n val STENCIL_CLEAR_VALUE: Int\n val STENCIL_FUNC: Int\n val
STENCIL_FAIL: Int\n val STENCIL_PASS_DEPTH_FAIL: Int\n val STENCIL_PASS_DEPTH_PASS:
Int\n val STENCIL_REF: Int\n val STENCIL_VALUE_MASK: Int\n val STENCIL_WRITEMASK:
Int\n val STENCIL_BACK_FUNC: Int\n val STENCIL_BACK_FAIL: Int\n
 val STENCIL_BACK_PASS_DEPTH_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_PASS: Int\n
val STENCIL_BACK_REF: Int\n val STENCIL_BACK_VALUE_MASK: Int\n val
STENCIL_BACK_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR_BOX: Int\n val
COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n
val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n
val SUBPIXEL_BITS: Int\n val RED_BITS: Int\n val GREEN_BITS: Int\n val BLUE_BITS: Int\n
val ALPHA_BITS: Int\n val DEPTH_BITS: Int\n val STENCIL_BITS: Int\n val

```

POLYGON\_OFFSET\_UNITS: Int\n val POLYGON\_OFFSET\_FACTOR: Int\n val  
 TEXTURE\_BINDING\_2D: Int\n val SAMPLE\_BUFFERS: Int\n val SAMPLES: Int\n val  
 SAMPLE\_COVERAGE\_VALUE: Int\n val SAMPLE\_COVERAGE\_INVERT: Int\n val  
 COMPRESSED\_TEXTURE\_FORMATS: Int\n val DONT\_CARE:  
 Int\n val FASTEST: Int\n val NICEST: Int\n val GENERATE\_MIPMAP\_HINT: Int\n val BYTE:  
 Int\n val UNSIGNED\_BYTE: Int\n val SHORT: Int\n val UNSIGNED\_SHORT: Int\n val INT:  
 Int\n val UNSIGNED\_INT: Int\n val FLOAT: Int\n val DEPTH\_COMPONENT: Int\n val  
 ALPHA: Int\n val RGB: Int\n val RGBA: Int\n val LUMINANCE: Int\n val  
 LUMINANCE\_ALPHA: Int\n val UNSIGNED\_SHORT\_4\_4\_4\_4: Int\n val  
 UNSIGNED\_SHORT\_5\_5\_5\_1: Int\n val UNSIGNED\_SHORT\_5\_6\_5: Int\n val  
 FRAGMENT\_SHADER: Int\n val VERTEX\_SHADER: Int\n val MAX\_VERTEX\_ATTRIBS: Int\n  
 val MAX\_VERTEX\_UNIFORM\_VECTORS: Int\n val MAX\_VARYING\_VECTORS: Int\n val  
 MAX\_COMBINED\_TEXTURE\_IMAGE\_UNITS: Int\n val MAX\_VERTEX\_TEXTURE\_IMAGE\_UNITS:  
 Int\n val MAX\_TEXTURE\_IMAGE\_UNITS: Int\n val MAX\_FRAGMENT\_UNIFORM\_VECTORS: Int\n  
 val SHADER\_TYPE: Int\n val DELETE\_STATUS:  
 Int\n val LINK\_STATUS: Int\n val VALIDATE\_STATUS: Int\n val ATTACHED\_SHADERS: Int\n  
 val ACTIVE\_UNIFORMS: Int\n val ACTIVE\_ATTRIBUTES: Int\n val  
 SHADING\_LANGUAGE\_VERSION: Int\n val CURRENT\_PROGRAM: Int\n val NEVER: Int\n val  
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n  
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n  
 val DECR: Int\n val INVERT: Int\n val INCR\_WRAP: Int\n val DECR\_WRAP: Int\n val  
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR:  
 Int\n val NEAREST\_MIPMAP\_NEAREST: Int\n val LINEAR\_MIPMAP\_NEAREST: Int\n val  
 NEAREST\_MIPMAP\_LINEAR: Int\n val LINEAR\_MIPMAP\_LINEAR: Int\n val  
 TEXTURE\_MAG\_FILTER: Int\n val TEXTURE\_MIN\_FILTER: Int\n  
 val TEXTURE\_WRAP\_S: Int\n val TEXTURE\_WRAP\_T: Int\n val TEXTURE\_2D: Int\n val  
 TEXTURE: Int\n val TEXTURE\_CUBE\_MAP: Int\n val TEXTURE\_BINDING\_CUBE\_MAP: Int\n  
 val TEXTURE\_CUBE\_MAP\_POSITIVE\_X: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_X: Int\n val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_Y: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Y: Int\n val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_Z: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Z: Int\n val  
 MAX\_CUBE\_MAP\_TEXTURE\_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val  
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val  
 TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val  
 TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val  
 TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17:  
 Int\n val TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21:  
 Int\n val TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25:  
 Int\n val TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29:  
 Int\n val TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE\_TEXTURE: Int\n val  
 REPEAT: Int\n val CLAMP\_TO\_EDGE: Int\n val MIRRORED\_REPEAT: Int\n val FLOAT\_VEC2:  
 Int\n val FLOAT\_VEC3: Int\n val FLOAT\_VEC4: Int\n val INT\_VEC2: Int\n val INT\_VEC3:  
 Int\n val INT\_VEC4: Int\n val BOOL: Int\n val BOOL\_VEC2: Int\n val BOOL\_VEC3: Int\n  
 val BOOL\_VEC4: Int\n val FLOAT\_MAT2: Int\n val FLOAT\_MAT3: Int\n val FLOAT\_MAT4: Int\n  
 val SAMPLER\_2D: Int\n val SAMPLER\_CUBE: Int\n val VERTEX\_ATTRIB\_ARRAY\_ENABLED:  
 Int\n  
 val VERTEX\_ATTRIB\_ARRAY\_SIZE: Int\n val VERTEX\_ATTRIB\_ARRAY\_STRIDE: Int\n val  
 VERTEX\_ATTRIB\_ARRAY\_TYPE: Int\n val VERTEX\_ATTRIB\_ARRAY\_NORMALIZED: Int\n val  
 VERTEX\_ATTRIB\_ARRAY\_POINTER: Int\n val VERTEX\_ATTRIB\_ARRAY\_BUFFER\_BINDING: Int\n



Int\n val CCW: Int\n val LINE\_WIDTH: Int\n val ALIASED\_POINT\_SIZE\_RANGE: Int\n val  
 ALIASED\_LINE\_WIDTH\_RANGE: Int\n val CULL\_FACE\_MODE: Int\n val FRONT\_FACE: Int\n  
 val DEPTH\_RANGE: Int\n val DEPTH\_WRITEMASK: Int\n val DEPTH\_CLEAR\_VALUE: Int\n val  
 DEPTH\_FUNC: Int\n val STENCIL\_CLEAR\_VALUE: Int\n val STENCIL\_FUNC: Int\n val  
 STENCIL\_FAIL:  
 Int\n val STENCIL\_PASS\_DEPTH\_FAIL: Int\n val STENCIL\_PASS\_DEPTH\_PASS: Int\n val  
 STENCIL\_REF: Int\n val STENCIL\_VALUE\_MASK: Int\n val STENCIL\_WRITEMASK: Int\n val  
 STENCIL\_BACK\_FUNC: Int\n val STENCIL\_BACK\_FAIL: Int\n val  
 STENCIL\_BACK\_PASS\_DEPTH\_FAIL: Int\n val STENCIL\_BACK\_PASS\_DEPTH\_PASS: Int\n val  
 STENCIL\_BACK\_REF: Int\n val STENCIL\_BACK\_VALUE\_MASK: Int\n val  
 STENCIL\_BACK\_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR\_BOX: Int\n val  
 COLOR\_CLEAR\_VALUE: Int\n val COLOR\_WRITEMASK: Int\n val UNPACK\_ALIGNMENT: Int\n  
 val PACK\_ALIGNMENT: Int\n val MAX\_TEXTURE\_SIZE: Int\n val MAX\_VIEWPORT\_DIMS: Int\n  
 val SUBPIXEL\_BITS: Int\n val RED\_BITS: Int\n val GREEN\_BITS: Int\n val BLUE\_BITS: Int\n  
 val ALPHA\_BITS: Int\n val DEPTH\_BITS: Int\n val STENCIL\_BITS: Int\n val  
 POLYGON\_OFFSET\_UNITS: Int\n  
 val POLYGON\_OFFSET\_FACTOR: Int\n val TEXTURE\_BINDING\_2D: Int\n val  
 SAMPLE\_BUFFERS: Int\n val SAMPLES: Int\n val SAMPLE\_COVERAGE\_VALUE: Int\n val  
 SAMPLE\_COVERAGE\_INVERT: Int\n val COMPRESSED\_TEXTURE\_FORMATS: Int\n val  
 DONT\_CARE: Int\n val FASTEST: Int\n val NICEST: Int\n val GENERATE\_MIPMAP\_HINT: Int\n  
 val BYTE: Int\n val UNSIGNED\_BYTE: Int\n val SHORT: Int\n val UNSIGNED\_SHORT: Int\n  
 val INT: Int\n val UNSIGNED\_INT: Int\n val FLOAT: Int\n val DEPTH\_COMPONENT: Int\n  
 val ALPHA: Int\n val RGB: Int\n val RGBA: Int\n val LUMINANCE: Int\n val  
 LUMINANCE\_ALPHA: Int\n val UNSIGNED\_SHORT\_4\_4\_4\_4: Int\n val  
 UNSIGNED\_SHORT\_5\_5\_5\_1: Int\n val UNSIGNED\_SHORT\_5\_6\_5: Int\n val  
 FRAGMENT\_SHADER: Int\n val VERTEX\_SHADER: Int\n val MAX\_VERTEX\_ATTRIBS: Int\n  
 val MAX\_VERTEX\_UNIFORM\_VECTORS: Int\n  
 val MAX\_VARYING\_VECTORS: Int\n val MAX\_COMBINED\_TEXTURE\_IMAGE\_UNITS: Int\n  
 val MAX\_VERTEX\_TEXTURE\_IMAGE\_UNITS: Int\n val MAX\_TEXTURE\_IMAGE\_UNITS: Int\n val  
 MAX\_FRAGMENT\_UNIFORM\_VECTORS: Int\n val SHADER\_TYPE: Int\n val DELETE\_STATUS:  
 Int\n val LINK\_STATUS: Int\n val VALIDATE\_STATUS: Int\n val ATTACHED\_SHADERS: Int\n  
 val ACTIVE\_UNIFORMS: Int\n val ACTIVE\_ATTRIBUTES: Int\n val  
 SHADING\_LANGUAGE\_VERSION: Int\n val CURRENT\_PROGRAM: Int\n val NEVER: Int\n val  
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n  
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n  
 val DECR: Int\n val INVERT: Int\n val INCR\_WRAP: Int\n val DECR\_WRAP: Int\n val  
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val  
 NEAREST: Int\n val LINEAR: Int\n val NEAREST\_MIPMAP\_NEAREST: Int\n val  
 LINEAR\_MIPMAP\_NEAREST: Int\n val NEAREST\_MIPMAP\_LINEAR: Int\n val  
 LINEAR\_MIPMAP\_LINEAR: Int\n val TEXTURE\_MAG\_FILTER: Int\n val TEXTURE\_MIN\_FILTER:  
 Int\n val TEXTURE\_WRAP\_S: Int\n val TEXTURE\_WRAP\_T: Int\n val TEXTURE\_2D: Int\n  
 val TEXTURE: Int\n val TEXTURE\_CUBE\_MAP: Int\n val TEXTURE\_BINDING\_CUBE\_MAP: Int\n  
 val TEXTURE\_CUBE\_MAP\_POSITIVE\_X: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_X: Int\n  
 val TEXTURE\_CUBE\_MAP\_POSITIVE\_Y: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Y: Int\n val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_Z: Int\n val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Z: Int\n val  
 MAX\_CUBE\_MAP\_TEXTURE\_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val  
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val  
 TEXTURE6: Int\n val TEXTURE7:

```

Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val TEXTURE10: Int\n val TEXTURE11:
Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val TEXTURE14: Int\n val TEXTURE15:
Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val TEXTURE18: Int\n val TEXTURE19:
Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val TEXTURE22: Int\n val TEXTURE23:
Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val TEXTURE26: Int\n val TEXTURE27:
Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val TEXTURE30: Int\n val TEXTURE31:
Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n val CLAMP_TO_EDGE: Int\n val
MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val FLOAT_VEC3: Int\n val FLOAT_VEC4:
Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val INT_VEC4: Int\n val BOOL: Int\n
 val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n
 val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n val SAMPLER_2D: Int\n val
SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED: Int\n val
VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n val
VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n val
VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int\n
 val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val
IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val
LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n
 val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val RENDERBUFFER:
Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val
DEPTH_COMPONENT16: Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val
DEPTH_STENCIL: Int\n val RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n
 val RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val
RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val
RENDERBUFFER_ALPHA_SIZE: Int\n val RENDERBUFFER_DEPTH_SIZE: Int\n val
RENDERBUFFER_STENCIL_SIZE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n
 val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n val
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n val
FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n val COLOR_ATTACHMENT0:
Int\n val DEPTH_ATTACHMENT: Int\n val STENCIL_ATTACHMENT: Int\n val
DEPTH_STENCIL_ATTACHMENT: Int\n val NONE: Int\n val FRAMEBUFFER_COMPLETE: Int\n
 val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n val
FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT:
Int\n val FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n val
FRAMEBUFFER_UNSUPPORTED: Int\n val FRAMEBUFFER_BINDING: Int\n val
RENDERBUFFER_BINDING: Int\n val MAX_RENDERBUFFER_SIZE: Int\n val
INVALID_FRAMEBUFFER_OPERATION: Int\n val UNPACK_FLIP_Y_WEBGL: Int\n val
UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n val CONTEXT_LOST_WEBGL: Int\n val
UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n val BROWSER_DEFAULT_WEBGL: Int\n
}\n}\n\n/*\n * Exposes the JavaScript
[WebGLContextEvent](https://developer.mozilla.org/en/docs/Web/API/WebGLContextEvent) to Kotlin\n
*/\n\npublic external open class WebGLContextEvent(type: String, eventInit: WebGLContextEventInit =
definedExternally) : Event {\n open val statusMessage: String\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external
interface WebGLContextEventInit : EventInit {\n var statusMessage: String? /* = \"\" */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun

```

```

WebGLContextEventInit(statusMessage: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): WebGLContextEventInit {
 val o = js("{}")
 o["statusMessage"] =
statusMessage
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
return o
}

* Exposes the JavaScript
[ArrayBuffer](https://developer.mozilla.org/en/docs/Web/API/ArrayBuffer) to Kotlin
public external open
class ArrayBuffer(length: Int) : BufferDataSource {
 open val byteLength: Int
 fun slice(begin: Int, end: Int =
definedExternally): ArrayBuffer
 companion object {
 fun isView(value: Any?): Boolean
 }
}

* Exposes the JavaScript
[ArrayBufferView](https://developer.mozilla.org/en/docs/Web/API/ArrayBufferView) to
Kotlin
public external interface ArrayBufferView : BufferDataSource {
 val buffer: ArrayBuffer
 val
byteOffset: Int
 val byteLength: Int
}

* Exposes the JavaScript
[Int8Array](https://developer.mozilla.org/en/docs/Web/API/Int8Array) to Kotlin
public external open class
Int8Array : ArrayBufferView {
 constructor(length: Int)
 constructor(array: Int8Array)
 constructor(array:
Array<Byte>)
 constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int =
definedExternally)
 open val length: Int
 override val buffer: ArrayBuffer
 override val byteOffset: Int
 override val byteLength: Int
 fun set(array: Int8Array, offset: Int = definedExternally)
 fun set(array:
Array<Byte>, offset: Int = definedExternally)
 fun subarray(start: Int, end: Int): Int8Array
 companion
object {
 val BYTES_PER_ELEMENT: Int
 }
}

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Int8Array.get(index: Int):
Byte = asDynamic()[index]

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Int8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }

* Exposes the JavaScript
[Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin
public external open class
Uint8Array : ArrayBufferView {
 constructor(length: Int)
 constructor(array: Uint8Array)
 constructor(array: Array<Byte>)
 constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)
 open val length: Int
 override val buffer: ArrayBuffer
 override val byteOffset:
Int
 override val byteLength: Int
 fun set(array: Uint8Array, offset: Int = definedExternally)
 fun
set(array: Array<Byte>, offset: Int = definedExternally)
 fun subarray(start: Int, end: Int): Uint8Array
 companion object {
 val BYTES_PER_ELEMENT: Int
 }
}

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.get(index: Int):
Byte = asDynamic()[index]

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }

* Exposes the JavaScript
[Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin
public external open class Uint8ClampedArray : ArrayBufferView {
 constructor(length: Int)
 constructor(array:
Uint8ClampedArray)
 constructor(array: Array<Byte>)
 constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)
 open val length: Int
 override val
buffer: ArrayBuffer
 override val byteOffset: Int
 override val byteLength: Int
 fun set(array:
Uint8ClampedArray, offset: Int = definedExternally)
 fun set(array: Array<Byte>, offset: Int =
definedExternally)
 fun subarray(start: Int, end: Int): Uint8ClampedArray
 companion object {
 val
BYTES_PER_ELEMENT: Int
 }
}

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun
Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun
Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }

* Exposes the JavaScript
[Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin
public external open class

```



```

Int16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int16Array)\n
constructor(array:
 Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int =
definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n
override val byteLength: Int\n fun set(array: Int16Array, offset: Int = definedExternally)\n fun set(array:
Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int16Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin\n */\n\npublic
external open class Uint16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array:
Uint16Array)\n constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n
override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint16Array, offset: Int =
definedExternally)\n fun set(array: Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end:
Int): Uint16Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.set(index: Int,
value: Short) { asDynamic()[index]
= value }\n\n/**\n * Exposes the JavaScript
[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin\n */\n\npublic external open class
Int32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int32Array)\n
constructor(array: Array<Int>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n
override val byteLength: Int\n fun set(array: Int32Array, offset: Int = definedExternally)\n fun set(array:
Array<Int>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int32Array\n\n companion object
{\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.get(index: Int): Int
= asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin\n */\n\npublic external open
class Uint32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint32Array)\n
constructor(array: Array<Int>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n
override val byteLength: Int\n fun set(array: Uint32Array, offset: Int = definedExternally)\n fun set(array:
Array<Int>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint32Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline operator fun Uint32Array.get(index: Int): Int =
asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin\n */\n\npublic external open

```

```

class Float32Array : ArrayBufferView {
 constructor(length: Int)
 constructor(array: Float32Array)
 constructor(array: Array<Float>)
 constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
 open val length: Int
 override val buffer: ArrayBuffer
 override val byteOffset: Int
 override val byteLength: Int
 fun set(array: Float32Array, offset: Int = definedExternally)
 fun set(array: Array<Float>, offset: Int = definedExternally)
 fun subarray(start: Int, end: Int): Float32Array

 companion object {
 val BYTES_PER_ELEMENT: Int
 }
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index]

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value }

/** Exposes the JavaScript [Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin */
public external open class Float64Array : ArrayBufferView {
 constructor(length: Int)
 constructor(array: Float64Array)
 constructor(array: Array<Double>)
 constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
 open val length: Int
 override val buffer: ArrayBuffer
 override val byteOffset: Int
 override val byteLength: Int
 fun set(array: Float64Array, offset: Int = definedExternally)
 fun set(array: Array<Double>, offset: Int = definedExternally)
 fun subarray(start: Int, end: Int): Float64Array

 companion object {
 val BYTES_PER_ELEMENT: Int
 }
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index]

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }

/** Exposes the JavaScript [DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin */
public external open class DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally) : ArrayBufferView {
 override val buffer: ArrayBuffer
 override val byteOffset: Int
 override val byteLength: Int
 fun getInt8(byteOffset: Int): Byte
 fun getUint8(byteOffset: Int): Byte
 fun getInt16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short
 fun getUint16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short
 fun getInt32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int
 fun getUint32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int
 fun getFloat32(byteOffset: Int, littleEndian: Boolean = definedExternally): Float
 fun getFloat64(byteOffset: Int, littleEndian: Boolean = definedExternally): Double
 fun setInt8(byteOffset: Int, value: Byte)
 fun setUint8(byteOffset: Int, value: Byte)
 fun setInt16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)
 fun setUint16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)
 fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)
 fun setUint32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)
 fun setFloat32(byteOffset: Int, value: Float, littleEndian: Boolean = definedExternally)
 fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean = definedExternally)
}

public external interface BufferDataSource
public external interface TexImageSource

/** Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details
package org.w3c.dom.clipboard
import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.dom.*
import org.w3c.dom.events.*

public external interface ClipboardEventInit : EventInit {
 val clipboardData: DataTransfer? /* = null */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public

```

```

inline fun ClipboardEventInit(clipboardData: DataTransfer? = null, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): ClipboardEventInit {\n val o = js("{}")\n
o["clipboardData"] = clipboardData\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n
o["composed"] = composed\n return o}\n\n/**\n * Exposes the JavaScript
[ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin\n */\npublic external
open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit = definedExternally) : Event {\n open
val clipboardData: DataTransfer?\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\n/**\n * Exposes the JavaScript [Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to
Kotlin\n */\npublic external
abstract class Clipboard : EventTarget {\n fun read(): Promise<DataTransfer>\n fun readText():
Promise<String>\n fun write(data: DataTransfer): Promise<Unit>\n fun writeText(data: String):
Promise<Unit>}\n\npublic external interface ClipboardPermissionDescriptor {\n var allowWithoutGesture:
Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally}\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
ClipboardPermissionDescriptor(allowWithoutGesture: Boolean? = false): ClipboardPermissionDescriptor {\n val
o = js("{}")\n o["allowWithoutGesture"] = allowWithoutGesture\n return o}\n", /*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat
for details\n\npackage org.w3c.dom.css\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport
org.w3c.dom.*\n\npublic external abstract class MediaList : ItemArrayLike<String> {\n open var mediaText:
String\n fun appendMedium(medium: String)\n fun deleteMedium(medium: String)\n override fun item(index:
Int): String?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun MediaList.get(index: Int):
String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin\n */\npublic external abstract
class StyleSheet {\n open val type: String\n open val href: String?\n open val ownerNode:
UnionElementOrProcessingInstruction?\n open val parentStyleSheet: StyleSheet?\n open val title: String?\n
open val media: MediaList\n open var disabled: Boolean\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet)
to Kotlin\n */\npublic external abstract class CSSStyleSheet : StyleSheet {\n open val ownerRule: CSSRule?\n
open val cssRules: CSSRuleList\n fun insertRule(rule: String, index: Int): Int\n fun deleteRule(index:
Int)\n}\n\n/**\n * Exposes the JavaScript
[StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin\n */\npublic external
abstract class StyleSheetList : ItemArrayLike<StyleSheet> {\n override fun item(index: Int):
StyleSheet?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun StyleSheetList.get(index: Int):
StyleSheet? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin\n */\npublic external interface
LinkStyle {\n val sheet: StyleSheet?\n get() = definedExternally}\n\n/**\n * Exposes the JavaScript
[CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList)
to Kotlin\n */\npublic external abstract class CSSRuleList : ItemArrayLike<CSSRule> {\n override fun
item(index: Int): CSSRule?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun CSSRuleList.get(index: Int):
CSSRule? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin\n */\npublic external abstract class

```

```

CSSRule {\n open val type: Short\n open var cssText: String\n open val parentRule: CSSRule?\n open val
parentStyleSheet: CSSStyleSheet?\n\n companion object {\n val STYLE_RULE: Short\n val
CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val
FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val
NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule)
to Kotlin\n */\npublic external abstract class CSSStyleRule : CSSRule {\n open var selectorText: String\n open
val style: CSSStyleDeclaration\n\n companion object {\n val STYLE_RULE: Short\n val
CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val
FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val
NAMESPACE_RULE: Short\n }\n}\n\npublic external abstract class CSSImportRule : CSSRule {\n open val
href: String\n open val media: MediaList\n open val styleSheet: CSSStyleSheet\n\n companion object {\n
val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val
MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val
MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule)
to Kotlin\n */\npublic external abstract class CSSGroupingRule : CSSRule {\n open val cssRules: CSSRuleList\n
fun insertRule(rule: String, index: Int): Int\n fun deleteRule(index: Int)\n\n companion object {\n val
STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val
MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val
MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin\n */\npublic external
abstract class CSSMediaRule : CSSGroupingRule {\n open val media: MediaList\n\n companion object {\n
val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val
MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n
val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin\n */\npublic external
abstract class CSSPageRule : CSSGroupingRule {\n open var selectorText: String\n open val style:
CSSStyleDeclaration\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE:
Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n
val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\npublic external
abstract class CSSMarginRule : CSSRule {\n open val name: String\n open val style:
CSSStyleDeclaration\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE:
Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n
val PAGE_RULE: Short\n val MARGIN_RULE:
Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin\n */\npublic
external abstract class CSSNamespaceRule : CSSRule {\n open val namespaceURI: String\n open val prefix:
String\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val
IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val
PAGE_RULE: Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n *
Exposes the JavaScript
[CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin\n */\npublic
external abstract class CSSStyleDeclaration : ItemArrayLike<String> {\n open var cssText: String\n open val
parentRule: CSSRule?\n open var cssFloat: String\n open var alignContent: String\n open var alignItems:
String\n open
var alignSelf: String\n open var animation: String\n open var animationDelay: String\n open var
animationDirection: String\n open var animationDuration: String\n open var animationFillMode: String\n open

```

var animationIterationCount: String\n open var animationName: String\n open var animationPlayState: String\n open var animationTimingFunction: String\n open var backfaceVisibility: String\n open var background: String\n open var backgroundAttachment: String\n open var backgroundClip: String\n open var backgroundColor: String\n open var backgroundImage: String\n open var backgroundOrigin: String\n open var backgroundPosition: String\n open var backgroundRepeat: String\n open var backgroundSize: String\n open var border: String\n open var borderBottom: String\n open var borderBottomColor: String\n open var borderBottomLeftRadius: String\n open var borderBottomRightRadius: String\n open var borderBottomStyle: String\n open var borderBottomWidth: String\n open var borderCollapse: String\n open var borderColor: String\n open var borderImage: String\n open var borderImageOutset: String\n open var borderImageRepeat: String\n open var borderImageSlice: String\n open var borderImageSource: String\n open var borderImageWidth: String\n open var borderLeft: String\n open var borderLeftColor: String\n open var borderLeftStyle: String\n open var borderLeftWidth: String\n open var borderRadius: String\n open var borderRight: String\n open var borderRightColor: String\n open var borderRightStyle: String\n open var borderRightWidth: String\n open var borderSpacing: String\n open var borderStyle: String\n open var borderTop: String\n open var borderTopColor: String\n open var borderTopLeftRadius: String\n open var borderTopRightRadius: String\n open var borderTopStyle: String\n open var borderTopWidth: String\n open var borderWidth: String\n open var bottom: String\n open var boxDecorationBreak: String\n open var boxShadow: String\n open var boxSizing: String\n open var breakAfter: String\n open var breakBefore: String\n open var breakInside: String\n open var captionSide: String\n open var clear: String\n open var clip: String\n open var color: String\n open var columnCount: String\n open var columnFill: String\n open var columnGap: String\n open var columnRule: String\n open var columnRuleColor: String\n open var columnRuleStyle: String\n open var columnRuleWidth: String\n open var columnSpan: String\n open var columnWidth: String\n open var columns: String\n open var content: String\n open var counterIncrement: String\n open var counterReset: String\n open var cursor: String\n open var direction: String\n open var display: String\n open var emptyCells: String\n open var filter: String\n open var flex: String\n open var flexBasis: String\n open var flexDirection: String\n open var flexFlow: String\n open var flexGrow: String\n open var flexShrink: String\n open var flexWrap: String\n open var font: String\n open var fontFamily: String\n open var fontFeatureSettings: String\n open var fontKerning: String\n open var fontLanguageOverride: String\n open var fontSize: String\n open var fontSizeAdjust: String\n open var fontStretch: String\n open var fontStyle: String\n open var fontSynthesis: String\n open var fontVariant: String\n open var fontVariantAlternates: String\n open var fontVariantCaps: String\n open var fontVariantEastAsian: String\n open var fontVariantLigatures: String\n open var fontVariantNumeric: String\n open var fontVariantPosition: String\n open var fontWeight: String\n open var hangingPunctuation: String\n open var height: String\n open var hyphens: String\n open var imageOrientation: String\n open var imageRendering: String\n open var imageResolution: String\n open var imeMode: String\n open var justifyContent: String\n open var left: String\n open var letterSpacing: String\n open var lineBreak: String\n open var lineHeight: String\n open var listStyle: String\n open var listStyleImage: String\n open var listStylePosition: String\n open var listStyleType: String\n open var margin: String\n open var marginBottom: String\n open var marginLeft: String\n open var marginRight: String\n open var marginTop: String\n open var mark: String\n open var markAfter: String\n open var markBefore: String\n open var marks: String\n open var marqueeDirection: String\n open var marqueePlayCount: String\n open var marqueeSpeed: String\n open var marqueeStyle: String\n open var mask: String\n open var maskType: String\n open var maxHeight: String\n open var maxWidth: String\n open var minHeight: String\n open var minWidth: String\n open var navDown: String\n open var navIndex: String\n open var navLeft: String\n open var navRight: String\n open var navUp: String\n open var objectFit: String\n open var objectPosition: String\n open var opacity: String\n open var order: String\n open var orphans: String\n open var outline: String\n open var outlineColor: String\n open var outlineOffset: String\n open var outlineStyle: String\n open

```

var outlineWidth: String\n open var overflowWrap: String\n open var overflowX: String\n open var
overflowY: String\n open var padding: String\n open var paddingBottom: String\n open var paddingLeft:
String\n open var paddingRight: String\n open var paddingTop: String\n open var pageBreakAfter: String\n
open var pageBreakBefore: String\n open var pageBreakInside: String\n open var perspective: String\n open
var perspectiveOrigin: String\n open var phonemes: String\n open var position:
String\n open var quotes: String\n open var resize: String\n open var rest: String\n open var restAfter:
String\n open var restBefore: String\n open var right: String\n open var tabSize: String\n open var
tableLayout: String\n open var textAlign: String\n open var textAlignLast: String\n open var
textCombineUpright: String\n open var textDecoration: String\n open var textDecorationColor: String\n open
var textDecorationLine: String\n open var textDecorationStyle: String\n open var textIndent: String\n open var
textJustify: String\n open var textOrientation: String\n open var textOverflow: String\n open var textShadow:
String\n open var textTransform: String\n open var textUnderlinePosition: String\n open var top: String\n
open var transform: String\n open var transformOrigin: String\n open var transformStyle: String\n open var
transition: String\n open var transitionDelay: String\n open var
transitionDuration: String\n open var transitionProperty: String\n open var transitionTimingFunction: String\n
open var unicodeBidi: String\n open var verticalAlign: String\n open var visibility: String\n open var
voiceBalance: String\n open var voiceDuration: String\n open var voicePitch: String\n open var
voicePitchRange: String\n open var voiceRate: String\n open var voiceStress: String\n open var voiceVolume:
String\n open var whiteSpace: String\n open var widows: String\n open var width: String\n open var
wordBreak: String\n open var wordSpacing: String\n open var wordWrap: String\n open var writingMode:
String\n open var zIndex: String\n open var _dashed_attribute: String\n open var _camel_cased_attribute:
String\n open var _webkit_cased_attribute: String\n fun getProperty(property: String): String\n fun
getPropertyPriority(property: String): String\n fun setProperty(property: String, value: String,
priority: String = definedExternally)\n fun setProperty(property: String, value: String)\n fun
setPropertyPriority(property: String, priority: String)\n fun removeProperty(property: String): String\n override
fun item(index: Int): String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]\n\n\npublic external interface
ElementCSSInlineStyle {\n val style: CSSStyleDeclaration\n}\n\n\n/**\n * Exposes the JavaScript
[CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin\n *\n\npublic external abstract class CSS {\n
companion object {\n fun escape(ident: String): String\n }\n}\n\n\npublic external interface
UnionElementOrProcessingInstruction, \"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in
the license/LICENSE.txt file.\n *\n\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n// See
github.com/kotlin/dukat for details\n\npackage org.w3c.dom.encryptedmedia\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n\n/**\n * Exposes the JavaScript
[MediaKeySystemConfiguration](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemConfiguration)
to Kotlin\n *\n\npublic external interface MediaKeySystemConfiguration {\n var label: String? /* = \"\" *\n
get() = definedExternally\n set(value) = definedExternally\n var initDataTypes: Array<String>? /* = arrayOf()
*\n
get() = definedExternally\n set(value) = definedExternally\n var audioCapabilities:
Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n
get() = definedExternally\n set(value) =
definedExternally\n var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n
get() = definedExternally\n
set(value) = definedExternally\n var distinctiveIdentifier: MediaKeysRequirement? /* =
MediaKeysRequirement.OPTIONAL *\n
get() = definedExternally\n set(value) = definedExternally\n
var persistentState: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL *\n
get() =
definedExternally\n set(value) = definedExternally\n var sessionTypes: Array<String>?\n
get() =
definedExternally\n set(value) = definedExternally\n }\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```

```

\`INVISIBLE_MEMBER\`)@kotlin.internal.InlineOnly\npublic inline fun MediaKeySystemConfiguration(label:
String? = \`\`, initDataTypes: Array<String>? = arrayOf(), audioCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), videoCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), distinctiveIdentifier: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, persistentState: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, sessionTypes: Array<String>?
= undefined): MediaKeySystemConfiguration {\n val o = js("{}")\n o["label"] = label\n
o["initDataTypes"] = initDataTypes\n o["audioCapabilities"] = audioCapabilities\n o["videoCapabilities"] =
videoCapabilities\n o["distinctiveIdentifier"] = distinctiveIdentifier\n o["persistentState"] = persistentState\n
o["sessionTypes"] = sessionTypes\n return o\n}\n\npublic external interface MediaKeySystemMediaCapability
{\n var contentType: String? /* = \`\` */\n get() = definedExternally\n set(value) = definedExternally\n
var robustness: String? /* = \`\` */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\`INVISIBLE_REFERENCE`,
\`INVISIBLE_MEMBER\`)@kotlin.internal.InlineOnly\npublic inline fun
MediaKeySystemMediaCapability(contentType: String? = \`\`, robustness: String? = \`\`):
MediaKeySystemMediaCapability {\n val o = js("{}")\n o["contentType"] = contentType\n
o["robustness"] = robustness\n return o\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySystemAccess](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemAccess) to Kotlin\n
*/\n\npublic external abstract class MediaKeySystemAccess {\n open val keySystem: String\n fun
getConfiguration(): MediaKeySystemConfiguration\n fun createMediaKeys(): Promise<MediaKeys>\n}\n\n/**\n
* Exposes the JavaScript [MediaKeys](https://developer.mozilla.org/en/docs/Web/API/MediaKeys) to Kotlin\n
*/\n\npublic external abstract class MediaKeys {\n fun createSession(sessionType: MediaKeySessionType =
definedExternally): MediaKeySession\n fun setServerCertificate(serverCertificate: dynamic):
Promise<Boolean>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySession](https://developer.mozilla.org/en/docs/Web/API/MediaKeySession) to Kotlin\n
*/\n\npublic external abstract class MediaKeySession : EventTarget {\n open val sessionId: String\n open val expiration:
Double\n open val closed:
Promise<Unit>\n open val keyStatuses: MediaKeyStatusMap\n open var onkeystatuschange: ((Event) ->
dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun generateRequest(initDataType: String,
initData: dynamic): Promise<Unit>\n fun load(sessionId: String): Promise<Boolean>\n fun update(response:
dynamic): Promise<Unit>\n fun close(): Promise<Unit>\n fun remove(): Promise<Unit>\n}\n\n/**\n * Exposes
the JavaScript [MediaKeyStatusMap](https://developer.mozilla.org/en/docs/Web/API/MediaKeyStatusMap) to
Kotlin\n
*/\n\npublic external abstract class MediaKeyStatusMap {\n open val size: Int\n fun has(keyId: dynamic):
Boolean\n fun get(keyId: dynamic): Any?\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyMessageEvent](https://developer.mozilla.org/en/docs/Web/API/MediaKeyMessageEvent) to Kotlin\n
*/\n\npublic external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) :
Event {\n open val messageType: MediaKeyMessageType\n
open val message: ArrayBuffer\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface MediaKeyMessageEventInit : EventInit {\n var messageType:
MediaKeyMessageType?\n var message: ArrayBuffer?\n}\n\n@Suppress(\`INVISIBLE_REFERENCE`,
\`INVISIBLE_MEMBER\`)@kotlin.internal.InlineOnly\npublic inline fun
MediaKeyMessageEventInit(messageType: MediaKeyMessageType?, message: ArrayBuffer?, bubbles: Boolean? =
false, cancelable: Boolean? = false, composed: Boolean? = false): MediaKeyMessageEventInit {\n val o =
js("{}")\n o["messageType"] = messageType\n o["message"] = message\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\npublic external open class
MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit = definedExternally) : Event {\n

```

```

open val initDataType: String\n open val initData: ArrayBuffer?\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface MediaEncryptedEventInit : EventInit {\n var initDataType: String? /* =
\"\" */\n get() = definedExternally\n set(value) = definedExternally\n var initData: ArrayBuffer? /* = null
*/\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaEncryptedEventInit(initDataType: String? = \"\", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): MediaEncryptedEventInit {\n val o = js(\"({})\")\n o[\"initDataType\"] = initDataType\n o[\"initData\"] = initData\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"]
=
cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeysRequirement {\n companion object\n}\n\npublic inline val
MediaKeysRequirement.Companion.REQUIRED: MediaKeysRequirement get() =
\"required\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.OPTIONAL: MediaKeysRequirement get() =
\"optional\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.NOT_ALLOWED: MediaKeysRequirement get() = \"not-
allowed\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeySessionType {\n companion object\n}\n\npublic inline val
MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() =
\"temporary\".asDynamic().unsafeCast<MediaKeySessionType>()\n\npublic
inline val MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() =
\"persistent-license\".asDynamic().unsafeCast<MediaKeySessionType>()\n\n/* please, don't implement this
interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface MediaKeyStatus {\n companion object\n}\n\npublic inline val
MediaKeyStatus.Companion.USABLE: MediaKeyStatus get() =
\"usable\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() =
\"expired\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() =
\"released\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = \"output-
restricted\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_DOWNSCALED:
MediaKeyStatus get() = \"output-downscaled\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = \"status-
pending\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = \"internal-
error\".asDynamic().unsafeCast<MediaKeyStatus>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeyMessageType {\n companion object\n}\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = \"license-
request\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = \"license-
renewal\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val

```



```

MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType
 get() = \"license-release\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.INDIVIDUALIZATION_REQUEST: MediaKeyMessageType get() =
\"individualization-request\".asDynamic().unsafeCast<MediaKeyMessageType>()\", /*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\/\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom.events\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\n\n/**\n * Exposes
the JavaScript [UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin\n *\n\npublic external
open class UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {\n open val view:
Window?\n open val detail: Int\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n\n\n\npublic external interface UIEventInit : EventInit {\n var view: Window? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var detail: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun UIEventInit(view: Window? = null,
detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): UIEventInit
{\n val o = js(\"({})\")\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin\n *\n\npublic external open class
FocusEvent(type:
String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {\n open val relatedTarget:
EventTarget?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n\n\n\npublic external interface FocusEventInit :
UIEventInit {\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun FocusEventInit(relatedTarget:
EventTarget? = null, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? =
false, composed: Boolean? = false): FocusEventInit {\n val o = js(\"({})\")\n o[\"relatedTarget\"] =
relatedTarget\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] =
cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent) to Kotlin\n *\n\npublic external open class MouseEvent(type: String, eventInitDict: MouseEventInit = definedExternally) :
UIEvent, UnionElementOrMouseEvent {\n open val screenX: Int\n open val screenY: Int\n open val clientX:
Int\n open val clientY: Int\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n open val altKey:
Boolean\n open val metaKey: Boolean\n open val button: Short\n open val buttons: Short\n open val
relatedTarget: EventTarget?\n open val region: String?\n open val pageX: Double\n open val pageY: Double\n
open val x: Double\n open val y: Double\n open val offsetX: Double\n open val offsetY: Double\n fun
getModifierState(keyArg: String): Boolean\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n\n\n\n\npublic
external interface MouseEventInit : EventModifierInit {\n var screenX: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var screenY: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var clientX: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var clientY: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var button: Short? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var buttons: Short? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var relatedTarget: EventTarget? /* = null */\n get()

```

```

= definedExternally\n set(value) = definedExternally\n var region: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MouseEventInit(screenX: Int? = 0,
screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget:
EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean?
= false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false,
modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false,
modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false,
modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MouseEventInit {\n val o =
js(\"({})\")\n o[\"screenX\"] = screenX\n o[\"screenY\"] = screenY\n o[\"clientX\"] = clientX\n o[\"clientY\"]
= clientY\n
 o[\"button\"] = button\n o[\"buttons\"] = buttons\n o[\"relatedTarget\"] = relatedTarget\n o[\"region\"] =
region\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] =
metaKey\n o[\"modifierAltGraph\"] = modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n
o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] =
modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] =
modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n
o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n
o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return
o\n}\n\npublic external interface EventModifierInit : UIEventInit {\n var ctrlKey: Boolean? /* = false */\n
get() = definedExternally\n
 set(value) = definedExternally\n var shiftKey: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var altKey: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var metaKey: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierAltGraph: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierCapsLock: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierFn: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierFnLock: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierHyper: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n
 var modifierNumLock: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var modifierScrollLock: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierSuper: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierSymbol: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierSymbolLock: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventModifierInit(ctrlKey: Boolean? =
false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean?
= false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? =
false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false,
view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): EventModifierInit {\n val o = js(\"({})\")\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] =
shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"] =
modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n
o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] =
modifierNumLock\n o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n

```

```

o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] =
view\n o["detail"]
= detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript [WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent)
to Kotlin\n *\npublic external open class WheelEvent(type: String, eventInitDict: WheelEventInit =
definedExternally) : MouseEvent {\n open val deltaX: Double\n open val deltaY: Double\n open val deltaZ:
Double\n open val deltaMode: Int\n\n companion object {\n val DOM_DELTA_PIXEL: Int\n val
DOM_DELTA_LINE: Int\n val DOM_DELTA_PAGE: Int\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface WheelEventInit : MouseEventInit {\n var deltaX: Double? /* = 0.0 */\n get()
= definedExternally\n set(value) = definedExternally\n var deltaY: Double? /* = 0.0 */\n get() =
definedExternally\n
 set(value) = definedExternally\n var deltaZ: Double? /* = 0.0 */\n get() = definedExternally\n
 set(value) = definedExternally\n var deltaMode: Int? /* = 0 */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun WheelEventInit(deltaX: Double? = 0.0,
deltaY: Double? = 0.0, deltaZ: Double? = 0.0, deltaMode: Int? = 0, screenX: Int? = 0, screenY: Int? = 0, clientX:
Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region:
String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean?
= false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false,
modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false,
modifierScrollLock: Boolean? = false,
 modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): WheelEventInit {\n val o = js("{}")\n o["deltaX"] = deltaX\n o["deltaY"] = deltaY\n
o["deltaZ"] = deltaZ\n o["deltaMode"] = deltaMode\n o["screenX"] = screenX\n o["screenY"] =
screenY\n o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n o["buttons"] =
buttons\n o["relatedTarget"] = relatedTarget\n o["region"] = region\n o["ctrlKey"] = ctrlKey\n
o["shiftKey"] = shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"] =
modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] = modifierHyper\n o["modifierNumLock"]
= modifierNumLock\n o["modifierScrollLock"] = modifierScrollLock\n o["modifierSuper"] =
modifierSuper\n o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] =
modifierSymbolLock\n o["view"] = view\n o["detail"] = detail\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to Kotlin\n *\npublic external open class
InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent {\n open val data: String\n
open val isComposing: Boolean\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface InputEventInit : UIEventInit {\n var data: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n
 var isComposing: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun InputEventInit(data: String? = "",
isComposing: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): InputEventInit {\n val o = js("{}")\n o["data"] = data\n
o["isComposing"] = isComposing\n o["view"] = view\n o["detail"] = detail\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript

```

```

[KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin\n *\npublic external
open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {\n
open val key: String\n open val
code: String\n open val location: Int\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n open val
altKey: Boolean\n open val metaKey: Boolean\n open val repeat: Boolean\n open val isComposing: Boolean\n
open val charCode: Int\n open val keyCode: Int\n open val which: Int\n fun getModifierState(keyArg:
String): Boolean\n\n companion object {\n val DOM_KEY_LOCATION_STANDARD: Int\n val
DOM_KEY_LOCATION_LEFT: Int\n val DOM_KEY_LOCATION_RIGHT: Int\n val
DOM_KEY_LOCATION_NUMPAD: Int\n val NONE: Short\n val CAPTURING_PHASE: Short\n
val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
KeyboardEventInit : EventModifierInit {\n var key: String? /* = \"\" */\n get() = definedExternally\n
set(value) = definedExternally\n var code: String? /* = \"\" */\n get() = definedExternally\n set(value) =
definedExternally\n var location:
Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var repeat: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var isComposing: Boolean? /* =
false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun KeyboardEventInit(key: String? = \"\",
code: String? = \"\", location: Int? = 0, repeat: Boolean? = false, isComposing: Boolean? = false, ctrlKey: Boolean?
= false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? =
false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock:
Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): KeyboardEventInit {\n val o = js(\"({})\")\n o[\"key\"] = key\n o[\"code\"] =
code\n o[\"location\"] = location\n o[\"repeat\"] = repeat\n o[\"isComposing\"] = isComposing\n
o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n
o[\"modifierAltGraph\"] = modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n
o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] =
modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] =
modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n
o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n
o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin\n *\npublic
external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) :
UIEvent {\n open val data: String\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface CompositionEventInit : UIEventInit {\n var data: String? /* = \"\" */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun CompositionEventInit(data: String? =
\"\", view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CompositionEventInit
{\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] =
bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the
JavaScript [Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin\n *\npublic external open class
Event(type: String, eventInitDict: EventInit = definedExternally) {\n open val type: String\n open val target:
EventTarget?\n open val currentTarget: EventTarget?\n open val eventPhase: Short\n open val bubbles:

```

```

Boolean\n open val cancelable: Boolean\n open val defaultPrevented: Boolean\n open val composed:
Boolean\n open val isTrusted: Boolean\n open val timeStamp: Number\n fun composedPath():
Array<EventTarget>\n fun stopPropagation()\n fun stopImmediatePropagation()\n fun preventDefault()\n
fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean)\n\n companion object
{\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n\n * Exposes the JavaScript
[EventTarget](https://developer.mozilla.org/en/docs/Web/API/EventTarget) to Kotlin\n *^/npublic external abstract
class EventTarget {\n fun addEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally)\n fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally)\n fun removeEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally)\n fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally)\n fun dispatchEvent(event: Event): Boolean\n}\n\n * Exposes the JavaScript
[EventListener](https://developer.mozilla.org/en/docs/Web/API/EventListener) to Kotlin\n *^/npublic external
interface EventListener {\n fun handleEvent(event: Event)\n}\n *^/n
* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^/n\n// NOTE: THIS
FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.clipboard.*\nimport
org.w3c.dom.css.*\nimport org.w3c.dom.encryptedmedia.*\nimport org.w3c.dom.events.*\nimport
org.w3c.dom.mediacapture.*\nimport org.w3c.dom.mediasource.*\nimport org.w3c.dom.pointerevents.*\nimport
org.w3c.dom.svg.*\nimport org.w3c.fetch.*\nimport org.w3c.files.*\nimport org.w3c.performance.*\nimport
org.w3c.workers.*\nimport org.w3c.xhr.*\n\npublic external abstract class HTMLAllCollection {\n open val
length: Int\n fun item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?\n fun
namedItem(name: String): UnionElementOrHTMLCollection?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? = asDynamic()[name]\n\n *^/n *
Exposes the JavaScript
[HTMLFormControlsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLFormControlsCollection)
to Kotlin\n *^/npublic external abstract class HTMLFormControlsCollection : HTMLCollection\n\n *^/n * Exposes
the JavaScript [RadioNodeList](https://developer.mozilla.org/en/docs/Web/API/RadioNodeList) to Kotlin\n
 *^/npublic external abstract class RadioNodeList : NodeList, UnionElementOrRadioNodeList {\n open var value:
String\n}\n\n *^/n * Exposes the JavaScript
[HTMLOptionsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionsCollection) to Kotlin\n
 *^/npublic
external abstract class HTMLOptionsCollection : HTMLCollection {\n override var length: Int\n open var
selectedIndex: Int\n fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic =
definedExternally)\n fun remove(index: Int)\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n *^/n *
Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to
Kotlin\n *^/npublic external abstract class HTMLInputElement : Element, GlobalEventHandlers,
DocumentAndElementEventHandlers, ElementContentEditable, ElementCSSInlineStyle {\n open var title:
String\n open var lang: String\n open var translate: Boolean\n open var dir: String\n open val dataset:
DOMStringMap\n open var hidden: Boolean\n open var tabIndex: Int\n open var accessKey: String\n

```

```

open val accessKeyLabel: String\n open var draggable: Boolean\n open val dropzone: DOMTokenList\n
open var contextMenu: HTMLMenuElement?\n open var spellcheck: Boolean\n open var innerText: String\n
open val offsetParent: Element?\n open val offsetTop: Int\n open val offsetLeft: Int\n open val offsetWidth:
Int\n open val offsetHeight: Int\n fun click()\n fun focus()\n fun blur()\n fun forceSpellCheck()\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n**\n * Exposes the
JavaScript [HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to
Kotlin\n */\npublic external abstract class HTMLUnknownElement : HTMLElement {\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n}\n\n**\n * Exposes the
JavaScript [DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin\n
*/\npublic external abstract class DOMStringMap\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun DOMStringMap.get(name:
String): String? = asDynamic()[name]\n\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun DOMStringMap.set(name:
String, value: String) { asDynamic()[name] = value }\n\n}\n\n**\n * Exposes the JavaScript
[HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin\n */\npublic
external abstract class HTMLHtmlElement : HTMLElement {\n open var version: String\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n}\n\n**\n * Exposes the JavaScript
[HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin\n */\npublic
external abstract class HTMLHeadElement : HTMLElement {\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

```

```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin \n * \n public
external abstract class HTMLTitleElement : HTMLElement { \n open var text: String \n \n companion object
{ \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n
val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLBaseElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBaseElement) to Kotlin \n * \n public
external abstract class HTMLBaseElement : HTMLElement { \n open var href: String \n open var target:
String \n \n companion object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n
val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE:
Short \n val ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val
COMMENT_NODE: Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n
 val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin \n * \n public
external abstract class HTMLLinkElement : HTMLElement, LinkStyle { \n
 open var href: String \n open var crossOrigin: String? \n open var rel: String \n open var `as`:
RequestDestination \n open val relList: DOMTokenList \n open var media: String \n open var nonce: String \n
open var hreflang: String \n open var type: String \n open val sizes: DOMTokenList \n open var referrerPolicy:
String \n open var charset: String \n open var rev: String \n open var target: String \n open var scope: String \n
open var workerType: WorkerType \n \n companion object { \n val ELEMENT_NODE: Short \n val
ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val
ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED:
Short \n val DOCUMENT_POSITION_PRECEDING: Short \n val
DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin \n * \n public
external abstract class HTMLMetaElement : HTMLElement { \n open var name: String \n open var httpEquiv:
String \n open var content: String \n open var scheme: String \n \n companion object { \n val
ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:

```

```

Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin\n */\npublic
external abstract class HTMLStyleElement : HTMLElement, LinkStyle {\n open var media: String\n open var
nonce: String\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement) to Kotlin\n */\npublic
external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {\n open var text: String\n
 open var link: String\n open var vLink: String\n open var aLink: String\n open var bgColor: String\n
 open
var background: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHeadingElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadingElement) to Kotlin\n
*/\npublic external abstract class HTMLHeadingElement : HTMLElement {\n open var align: String\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLParagraphElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParagraphElement) to Kotlin\n
*/\npublic external abstract class HTMLParagraphElement : HTMLElement {\n open var align: String\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE:

```



```

Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHRElement) to Kotlin\n */\npublic
external abstract class HTMLHRElement : HTMLInputElement {\n open var align: String\n open var color: String\n
open var noShade: Boolean\n open var size: String\n open var width: String\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin\n */\npublic
external abstract class HTMLPreElement : HTMLInputElement {\n open var width: Int\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin\n */\npublic
external abstract class HTMLQuoteElement : HTMLInputElement {\n open var cite:
String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLOListElement) to Kotlin\n */\npublic
external abstract class HTMLLOListElement : HTMLInputElement {\n open
var reversed: Boolean\n open var start: Int\n open var type: String\n open var compact: Boolean\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val

```

```

COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement)
to Kotlin\n */\npublic external abstract class HTMLUListElement : HTMLInputElement {\n open var compact:
Boolean\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLIElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLIElement)
to Kotlin\n */\npublic external abstract class HTMLLIElement : HTMLInputElement {\n open var value: Int\n open
var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDListElement)
to Kotlin\n */\npublic external abstract class HTMLDListElement : HTMLInputElement {\n open var compact:
Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDivElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDivElement) to Kotlin\n */\npublic
external abstract class HTMLDivElement : HTMLInputElement {\n open var align: String\n\n companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

```

```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLAnchorElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAnchorElement) to Kotlin \n
*\npublic external abstract class HTMLAnchorElement : HTMLElement, HTMLHyperlinkElementUtils { \n open
var target: String \n open var download: String \n open var ping: String \n open var rel: String \n open val
relList: DOMTokenList \n open var hreflang: String \n open var type: String \n open var text: String \n open
var referrerPolicy: String \n open var coords: String \n open var charset: String \n open var name: String \n
open var rev: String \n open var shape: String \n \n companion object { \n val ELEMENT_NODE: Short \n
val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n
val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n
 val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin \n
*\npublic external abstract class HTMLDataElement : HTMLElement { \n open var value: String \n \n companion object { \n
 val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:
Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE:
Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin \n
*\npublic external abstract class HTMLTimeElement : HTMLElement { \n open var dateTime: String \n \n companion
object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE:
Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE:
Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin \n
*\npublic external abstract class HTMLSpanElement : HTMLElement { \n companion object { \n val
ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:
Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n

```

```

 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin \n * \n public
external abstract class HTMLBRElement : HTMLElement { \n open var clear: String \n \n companion object { \n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to
Kotlin \n * \n public external interface HTMLHyperlinkElementUtils { \n var href: String \n val origin: String \n
var protocol: String \n var username: String \n var password: String \n var host: String \n var hostname:
String \n var port: String \n var pathname: String \n var search: String \n var hash: String \n} \n \n /** \n * Exposes
the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to
Kotlin \n * \n public
external abstract class HTMLModElement : HTMLElement { \n open var cite: String \n open var dateTime:
String \n \n companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement)
to Kotlin \n * \n public external abstract class HTMLPictureElement : HTMLElement { \n companion object { \n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement)
to Kotlin \n * \n public external abstract class HTMLSourceElement : HTMLElement { \n open var src: String \n
open var type: String \n open var srcset: String \n open var sizes: String \n open var media: String \n \n
companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:

```

```

Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
 }\n}\n\n/**\n * Exposes the JavaScript
[HTMLImageElement](https://developer.mozilla.org/en/docs/Web/API/HTMLImageElement) to Kotlin\n
*/\npublic external abstract class HTMLImageElement : HTMLInputElement, HTMLImageElement,
TexImageSource {\n open var alt: String\n open var src: String\n open var srcset: String\n open var sizes:
String\n open var crossOrigin: String?\n open var useMap: String\n open var isMap: Boolean\n open var
width: Int\n open var height: Int\n open val naturalWidth: Int\n open val naturalHeight: Int\n open val
complete: Boolean\n open val currentSrc: String\n open var referrerPolicy: String\n open var name: String\n
open var lowsrc: String\n open var align: String\n open var hspace: Int\n open var vspace: Int\n open var
longDesc: String\n open var border: String\n open val x: Int\n open val y: Int\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin\n
*/\npublic external abstract class HTMLIFrameElement : HTMLInputElement {\n open var src: String\n open var
srcdoc: String\n open var name: String\n open val sandbox: DOMTokenList\n
 open var allowFullscreen: Boolean\n open var allowUserMedia: Boolean\n open var width: String\n open
var height: String\n open var referrerPolicy: String\n open val contentDocument: Document?\n open val
contentWindow: Window?\n open var align: String\n open var scrolling: String\n open var frameborder:
String\n open var longDesc: String\n open var marginHeight: String\n open var marginWidth: String\n fun
getSVGDocument(): Document?\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin\n
*/\npublic external abstract class HTMLEmbedElement : HTMLInputElement {\n open var src: String\n open var
type: String\n open var width: String\n open var height: String\n open var align: String\n open var name:
String\n fun getSVGDocument(): Document?\n\n companion object {\n val ELEMENT_NODE: Short\n

```

```

val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE:
Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n
val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement) to
Kotlin\n */\npublic external abstract class HTMLElement : HTMLInputElement {\n open var data: String\n
open var type: String\n open var typeMustMatch: Boolean\n open var name: String\n open var useMap:
String\n open val form: HTMLFormElement?\n open var width: String\n open var height: String\n open val
contentDocument: Document?\n open val contentWindow: Window?\n open val willValidate:
Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open var align: String\n
open var archive: String\n open var code: String\n open var declare: Boolean\n open var hspace: Int\n open
var standby: String\n open var vspace: Int\n open var codeBase: String\n open var codeType: String\n open
var border: String\n fun getSVGDocument(): Document?\n fun checkValidity(): Boolean\n fun
reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n
val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to
Kotlin\n */\npublic external abstract class HTMLParamElement : HTMLInputElement {\n open var name: String\n
open var value: String\n open var type: String\n open var valueType: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement) to Kotlin\n */\npublic
external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {\n open var
width: Int\n open var height: Int\n open val videoWidth: Int\n open val videoHeight: Int\n open var
poster: String\n open var playsInline: Boolean\n\n companion object {\n val NETWORK_EMPTY: Short\n
val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE:
Short\n val HAVE_NOTHING:
Short\n val HAVE_METADATA: Short\n val HAVE_CURRENT_DATA: Short\n val
HAVE_FUTURE_DATA: Short\n val HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n

```

```

 val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
 val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript
[HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement)
to Kotlin\n */\npublic external abstract class HTMLAudioElement : HTMLMediaElement {\n companion object
{\n val NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING:
Short\n val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val
HAVE_METADATA: Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA:
Short\n val HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the
JavaScript [HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin\n
*/\npublic external abstract class HTMLTrackElement : HTMLMediaElement {\n open var kind: String\n open var src:
String\n open var srclang: String\n open var label: String\n open var default: Boolean\n open val readyState:
Short\n open val track: TextTrack\n\n companion object {\n val NONE: Short\n val LOADING: Short\n
 val LOADED: Short\n val ERROR: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE:
Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val
DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val
NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes
the JavaScript [HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to
Kotlin\n */\npublic external abstract class HTMLMediaElement : HTMLMediaElement {\n open val error:
MediaError?\n open var src: String\n open var srcObject: MediaProvider?\n open val currentSrc: String\n
open var crossOrigin: String?\n open val networkState: Short\n open var preload: String\n open val buffered:
TimeRanges\n open val readyState:
Short\n open val seeking: Boolean\n open var currentTime: Double\n open val duration: Double\n open val
paused: Boolean\n open var defaultPlaybackRate: Double\n open var playbackRate: Double\n open val played:
TimeRanges\n open val seekable: TimeRanges\n open val ended: Boolean\n open var autoplay: Boolean\n
open var loop: Boolean\n open var controls: Boolean\n open var volume: Double\n open var muted: Boolean\n
open var defaultMuted: Boolean\n open val audioTracks: AudioTrackList\n open val videoTracks:

```

```

VideoTrackList\n open val textTracks: TextTrackList\n open val mediaKeys: MediaKeys?\n open var
onencrypted: ((Event) -> dynamic)?\n open var onwaitingforkey: ((Event) -> dynamic)?\n fun load()\n fun
canPlayType(type: String): CanPlayTypeResult\n fun fastSeek(time: Double)\n fun getStartDate(): dynamic\n
fun play(): Promise<Unit>\n fun pause()\n fun addTextTrack(kind: TextTrackKind, label: String
= definedExternally, language: String = definedExternally): TextTrack\n fun setMediaKeys(mediaKeys:
MediaKeys?): Promise<Unit>\n\n companion object {\n val NETWORK_EMPTY: Short\n val
NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE: Short\n
val HAVE_NOTHING: Short\n val HAVE_METADATA: Short\n val HAVE_CURRENT_DATA:
Short\n val HAVE_FUTURE_DATA: Short\n val HAVE_ENOUGH_DATA: Short\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n/**\n * Exposes the
JavaScript [MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin\n *\npublic external
abstract class MediaError {\n open val code: Short\n\n companion object {\n val
MEDIA_ERR_ABORTED: Short\n val MEDIA_ERR_NETWORK: Short\n val MEDIA_ERR_DECODE:
Short\n val MEDIA_ERR_SRC_NOT_SUPPORTED: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin\n *\npublic external
abstract class AudioTrackList : EventTarget {\n open val length: Int\n open var onchange: ((Event) ->
dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) ->
dynamic)?\n
fun getTrackById(id: String): AudioTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun AudioTrackList.get(index:
Int): AudioTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin\n *\npublic external abstract
class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n open val label: String\n open val language: String\n open var enabled: Boolean\n open val
sourceBuffer: SourceBuffer?\n}\n\n/**\n * Exposes the JavaScript
[VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to Kotlin\n *\npublic external
abstract class VideoTrackList : EventTarget {\n open val length: Int\n open val selectedIndex: Int\n open var
onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var
onremovetrack:
((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
VideoTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun VideoTrackList.get(index:
Int): VideoTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin\n *\npublic external abstract
class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n open val label: String\n open val language: String\n open var selected: Boolean\n open val
sourceBuffer: SourceBuffer?\n}\n\npublic external abstract class TextTrackList : EventTarget {\n open val length:
Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open
var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
TextTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",

```



```

\ "INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public
inline operator fun TextTrackList.get(index: Int): TextTrack? = asDynamic()[index] \n \n /** \n * Exposes the
JavaScript [TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin \n * \n public external
abstract class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack { \n open val kind:
TextTrackKind \n open val label: String \n open val language: String \n open val id: String \n open val
inBandMetadataTrackDispatchType: String \n open var mode: TextTrackMode \n open val cues:
TextTrackCueList? \n open val activeCues: TextTrackCueList? \n open var oncuechange: ((Event) ->
dynamic)? \n open val sourceBuffer: SourceBuffer? \n fun addCue(cue: TextTrackCue) \n fun removeCue(cue:
TextTrackCue) \n } \n \n public external abstract class TextTrackCueList { \n open val length: Int \n fun
getCueById(id: String): TextTrackCue? \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public
inline operator fun TextTrackCueList.get(index: Int): TextTrackCue? = asDynamic()[index] \n \n /** \n * Exposes the
JavaScript [TextTrackCue](https://developer.mozilla.org/en/docs/Web/API/TextTrackCue) to Kotlin \n * \n public
external abstract class TextTrackCue : EventTarget { \n open val track: TextTrack? \n open var id: String \n
 open var startTime: Double \n open var endTime: Double \n open var pauseOnExit: Boolean \n open var
onenter: ((Event) -> dynamic)? \n open var onexit: ((Event) -> dynamic)? \n } \n \n /** \n * Exposes the JavaScript
[TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin \n * \n public external abstract
class TimeRanges { \n open val length: Int \n fun start(index: Int): Double \n fun end(index: Int):
Double \n } \n \n /** \n * Exposes the JavaScript
[TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin \n * \n public external open class
TrackEvent(type: String, eventInitDict: TrackEventInit
= definedExternally) : Event { \n open val track: UnionAudioTrackOrTextTrackOrVideoTrack? \n companion
object { \n val NONE: Short \n val CAPTURING_PHASE: Short \n val AT_TARGET: Short \n val
BUBBLING_PHASE: Short \n } \n } \n \n public external interface TrackEventInit : EventInit { \n var track:
UnionAudioTrackOrTextTrackOrVideoTrack? /* = null */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun TrackEventInit(track:
UnionAudioTrackOrTextTrackOrVideoTrack? = null, bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): TrackEventInit { \n val o = js("{}") \n o["track"] = track \n o["bubbles"] =
bubbles \n o["cancelable"] = cancelable \n o["composed"] = composed \n return o \n } \n \n /** \n * Exposes the
JavaScript [HTMLMapElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement)
to Kotlin \n * \n public external abstract class HTMLMapElement : HTMLElement { \n open var name: String \n
 open val areas: HTMLCollection \n companion object { \n val ELEMENT_NODE: Short \n val
ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val
ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n val
DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n val
DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n } \n \n /** \n * Exposes the JavaScript
[HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement)
to Kotlin \n * \n public external abstract class HTMLAreaElement : HTMLElement, HTMLHyperlinkElementUtils
{ \n open var alt: String \n open var coords: String \n open var shape: String \n open var target: String \n
 open var download: String \n open var ping: String \n open var rel: String \n open val relList: DOMTokenList \n
 open var referrerPolicy: String \n open var noHref: Boolean \n companion object { \n val
ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val

```

```

CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}n/n/**n * Exposes the JavaScript
[HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin\n */npublic
external abstract class HTMLTableElement : HTMLElement {n open var caption:
HTMLTableCaptionElement?n open var tHead: HTMLTableSectionElement?n open var tFoot:
HTMLTableSectionElement?n open val tBodies: HTMLCollection\n open val rows: HTMLCollection\n open
var align: String\n open var border: String\n open var frame: String\n open var rules: String\n open var
summary: String\n open var width: String\n open var bgColor: String\n open var cellPadding: String\n open
var cellSpacing: String\n fun createCaption(): HTMLTableCaptionElement\n fun deleteCaption()\n
 fun createTHead(): HTMLTableSectionElement\n fun deleteTHead()\n fun createTFoot():
HTMLTableSectionElement\n fun deleteTFoot()\n fun createTBody(): HTMLTableSectionElement\n fun
insertRow(index: Int = definedExternally): HTMLTableRowElement\n fun deleteRow(index: Int)\n\n
companion object {n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}n/n/**n * Exposes
the JavaScript
[HTMLTableCaptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCaptionElement) to
Kotlin\n */npublic external abstract class HTMLTableCaptionElement : HTMLElement {n open var align:
String\n\n companion object {n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}n/n/**n * Exposes the JavaScript
[HTMLTableColElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableColElement) to Kotlin\n
*/npublic external abstract class HTMLTableColElement : HTMLElement {n open var span: Int\n open var
align: String\n open var ch: String\n open var chOff: String\n open var vAlign: String\n open var width:
String\n\n companion object {n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

```

```

DOCUMENT_POSITION_DISCONNECTED:
Short val DOCUMENT_POSITION_PRECEDING: Short val
DOCUMENT_POSITION_FOLLOWING: Short val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n \n /** * Exposes the JavaScript
[HTMLTableSectionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableSectionElement) to Kotlin
*\npublic external abstract class HTMLTableSectionElement : HTMLElement {
open val rows: HTMLCollection
open var align: String
open var ch: String
open var chOff: String
open var vAlign:
String
fun insertRow(index: Int = definedExternally): HTMLElement
fun deleteRow(index: Int)\n\n
companion object {
val ELEMENT_NODE: Short val ATTRIBUTE_NODE: Short val
TEXT_NODE: Short val CDATA_SECTION_NODE: Short val ENTITY_REFERENCE_NODE:
Short val ENTITY_NODE: Short val PROCESSING_INSTRUCTION_NODE:
Short val COMMENT_NODE: Short val DOCUMENT_NODE: Short val
DOCUMENT_TYPE_NODE: Short val DOCUMENT_FRAGMENT_NODE: Short val
NOTATION_NODE: Short val DOCUMENT_POSITION_DISCONNECTED: Short val
DOCUMENT_POSITION_PRECEDING: Short val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short val DOCUMENT_POSITION_CONTAINED_BY:
Short val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n \n /** * Exposes
the JavaScript
[HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin
*\npublic external abstract class HTMLTableRowElement : HTMLElement {
open val rowIndex: Int
open val sectionRowIndex: Int
open val cells: HTMLCollection
open var align: String
open var ch: String
open var chOff: String
open var vAlign: String
open var bgColor: String
fun insertCell(index: Int =
definedExternally):
HTMLElement
fun deleteCell(index: Int)\n\n
companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short val TEXT_NODE: Short val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short val ENTITY_NODE: Short val
PROCESSING_INSTRUCTION_NODE: Short val COMMENT_NODE: Short val
DOCUMENT_NODE: Short val DOCUMENT_TYPE_NODE: Short val
DOCUMENT_FRAGMENT_NODE: Short val NOTATION_NODE: Short val
DOCUMENT_POSITION_DISCONNECTED: Short val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n \n /** * Exposes the JavaScript
[HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin
*\npublic external
abstract class HTMLTableCellElement : HTMLElement {
open var colSpan: Int
open var rowSpan: Int
open var headers: String
open val cellIndex: Int
open var scope: String
open var abbr: String
open var align: String
open var axis: String
open var height: String
open var width: String
open var ch:
String
open var chOff: String
open var noWrap: Boolean
open var vAlign: String
open var bgColor:
String\n\n
companion object {
val ELEMENT_NODE: Short val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short val CDATA_SECTION_NODE: Short val ENTITY_REFERENCE_NODE:
Short val ENTITY_NODE: Short val PROCESSING_INSTRUCTION_NODE: Short val
COMMENT_NODE: Short val DOCUMENT_NODE: Short val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short val NOTATION_NODE: Short val
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short val DOCUMENT_POSITION_FOLLOWING:
Short
val DOCUMENT_POSITION_CONTAINS: Short val

```

```

DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin\n */\npublic
external abstract class HTMLFormElement : HTMLInputElement {\n open var acceptCharset: String\n open var
action: String\n open var autocomplete: String\n open var enctype: String\n open var encoding: String\n open
var method: String\n open var name: String\n open var noValidate: Boolean\n open var target: String\n open
val elements: HTMLFormControlsCollection\n open val length: Int\n fun submit()\n fun reset()\n fun
checkValidity(): Boolean\n fun reportValidity(): Boolean\n\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLFormElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]\n\n/**\n *
Exposes the JavaScript
[HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin\n */\npublic
external abstract class HTMLLabelElement : HTMLInputElement {\n open val form: HTMLFormElement?\n open
var htmlFor: String\n open val control: HTMLInputElement?\n\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to
Kotlin\n */\npublic external abstract class HTMLInputElement : HTMLInputElement {\n open var accept: String\n
open var alt: String\n open var autocomplete: String\n open var autofocus: Boolean\n open var
defaultChecked: Boolean\n open var checked: Boolean\n open var dirName: String\n open var disabled:
Boolean\n open val form: HTMLFormElement?\n open val files: FileList?\n open var formAction: String\n
open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open var
formTarget: String\n open var height: Int\n open var indeterminate: Boolean\n open var inputMode: String\n
open val list: HTMLInputElement?\n open var
max: String\n open var maxLength: Int\n open var min: String\n open var minLength: Int\n open var
multiple: Boolean\n open var name: String\n open var pattern: String\n open var placeholder: String\n open
var readOnly: Boolean\n open var required: Boolean\n open var size: Int\n open var src: String\n open var

```

```

step: String\n open var type: String\n open var defaultValue: String\n open var value: String\n open var
valueAsDate: dynamic\n open var valueAsNumber: Double\n open var width: Int\n open val willValidate:
Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n
 open var selectionStart: Int?\n open var selectionEnd: Int?\n open var selectionDirection: String?\n open var
align: String\n open var useMap: String\n fun stepUp(n: Int = definedExternally)\n fun stepDown(n: Int =
definedExternally)\n fun checkValidity(): Boolean\n fun reportValidity():
Boolean\n fun setCustomValidity(error: String)\n fun select()\n fun setRangeText(replacement: String)\n
fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)\n
fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to
Kotlin\n */\npublic external abstract class HTMLButtonElement : HTMLElement {\n open var autofocus:
Boolean\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var formAction: String\n
 open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open
var formTarget: String\n open var name: String\n open var type: String\n open var value: String\n open var
menu: HTMLMenuElement?\n open val willValidate: Boolean\n open val validity: ValidityState\n open val
validationMessage: String\n open val labels: NodeList\n fun checkValidity(): Boolean\n fun reportValidity():
Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin\n */\npublic
external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {\n open var
autocomplete: String\n
 open var autofocus: Boolean\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open
var multiple: Boolean\n open var name: String\n open var required: Boolean\n open var size: Int\n open val
type: String\n open val options: HTMLOptionsCollection\n override var length: Int\n open val
selectedOptions: HTMLCollection\n open var selectedIndex: Int\n open var value: String\n open val
willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val
labels: NodeList\n fun namedItem(name: String): HTMLOptionElement?\n fun add(element:
UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n fun
remove(index: Int)\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun
setCustomValidity(error: String)\n override fun item(index: Int): Element?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:

```

```

Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
HTMLSelectElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
HTMLSelectElement.set(index:
Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n * Exposes the JavaScript
[HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin\n
*\n\npublic external abstract class HTMLDataListElement : HTMLInputElement {\n open val options:
HTMLCollection\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin\n
*\n\npublic external abstract class HTMLOptGroupElement : HTMLInputElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n open var disabled: Boolean\n open var label:
String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin\n
*\n\npublic external abstract class HTMLOptionElement : HTMLInputElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n open var disabled: Boolean\n open val form:
HTMLFormElement?\n open var label: String\n open var defaultSelected: Boolean\n open var selected:
Boolean\n open var value: String\n open var text: String\n open val index: Int\n\n companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n

```

```

 val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin \n
*/ \n public external abstract class HTMLTextAreaElement : HTMLElement { \n open var autocomplete: String \n
open var autofocus: Boolean \n open var cols: Int \n open var dirName: String \n open var disabled: Boolean \n
open val form: HTMLFormElement? \n open var inputMode: String \n open var maxLength: Int \n open var
minLength: Int \n open var name:
String \n open var placeholder: String \n open var readOnly: Boolean \n open var required: Boolean \n open
var rows: Int \n open var wrap: String \n open val type: String \n open var defaultValue: String \n open var
value: String \n open val maxLength: Int \n open val willValidate: Boolean \n open val validity: ValidityState \n
open val validationMessage: String \n open val labels: NodeList \n open var selectionStart: Int? \n open var
selectionEnd: Int? \n open var selectionDirection: String? \n fun checkValidity(): Boolean \n fun
reportValidity(): Boolean \n fun setCustomValidity(error: String) \n fun select() \n fun
setRangeText(replacement: String) \n fun setRangeText(replacement: String, start: Int, end: Int, selectionMode:
SelectionMode = definedExternally) \n fun setSelectionRange(start: Int, end: Int, direction: String =
definedExternally) \n \n companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to Kotlin \n
*/ \n public external abstract class HTMLKeygenElement : HTMLElement { \n open var autofocus: Boolean \n
open var challenge: String \n open var disabled: Boolean \n open val
form: HTMLFormElement? \n open var keytype: String \n open var name: String \n open val type: String \n
open val willValidate: Boolean \n open val validity: ValidityState \n open val validationMessage: String \n open
val labels: NodeList \n fun checkValidity(): Boolean \n fun reportValidity(): Boolean \n fun
setCustomValidity(error: String) \n \n companion object { \n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin \n
*/ \n public external abstract class HTMLOutputElement : HTMLElement { \n open val htmlFor: DOMTokenList \n

```





```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to
Kotlin\n */\npublic external abstract class HTMLLegendElement : HTMLElement {\n open val form:
HTMLFormElement?\n open var align: String\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin\n */\npublic external
abstract class ValidityState {\n open val valueMissing: Boolean\n open val typeMismatch: Boolean\n open val
patternMismatch: Boolean\n open val tooLong: Boolean\n open val tooShort: Boolean\n open val
rangeUnderflow: Boolean\n open val rangeOverflow: Boolean\n open val stepMismatch: Boolean\n open val
badInput: Boolean\n open val customError: Boolean\n open val valid: Boolean\n}\n}\n\n/**\n * Exposes the
JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to
Kotlin\n */\npublic external
abstract class HTMLDetailsElement : HTMLElement {\n open var open: Boolean\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
HTMLMenuElement : HTMLElement {\n open var type: String\n open var label: String\n open var
compact: Boolean\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
HTMLMenuItemElement : HTMLElement {\n open var type: String\n open var label: String\n open var icon:

```

```

String\n open var disabled: Boolean\n open var
checked: Boolean\n open var radiogroup: String\n open var default: Boolean\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n public external open class
RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {\n open val
relatedTarget: EventTarget?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n\n public external interface
RelatedEventInit : EventInit {\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n
set(value) = definedExternally\n }\n\n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n @kotlin.internal.InlineOnly\n public inline fun RelatedEventInit(relatedTarget:
EventTarget? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
RelatedEventInit {\n val o = js("{}")\n o["relatedTarget"] = relatedTarget\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n }\n\n /**\n * Exposes the JavaScript
[HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin\n
*/\n public external abstract class HTMLDialogElement
: HTMLElement {\n open var open: Boolean\n open var returnValue: String\n fun show(anchor:
UnionElementOrMouseEvent = definedExternally)\n fun showModal(anchor: UnionElementOrMouseEvent =
definedExternally)\n fun close(returnValue: String = definedExternally)\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n /**\n * Exposes the
JavaScript [HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin\n
*/\n public external abstract class HTMLScriptElement : HTMLElement, HTMLScriptElement {\n open
var src: String\n open var type: String\n open var charset: String\n open var async: Boolean\n open var defer:
Boolean\n open var crossOrigin: String?\n open var text: String\n open var nonce: String\n open var event:
String\n open var htmlFor: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE:
Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
 }\n\n /**\n * Exposes

```

the JavaScript [HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin

```

public external abstract class HTMLTemplateElement : HTMLElement {
 open val content: DocumentFragment
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

public external interface AssignedNodesOptions {
 var flatten: Boolean? /* = false */
 fun get(): Array<Node>
 fun set(value): Boolean? /* = false */
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): Array<Node> {
 val o = js("{}")
 o["flatten"] = flatten
 return o
}

public external abstract class HTMLSlotElement : HTMLElement {
 open var name: String
 fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

public external interface AssignedNodesOptions {
 var flatten: Boolean? /* = false */
 fun get(): Boolean? /* = false */
 fun set(value): Boolean? /* = false */
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): Boolean? {
 val o = js("{}")
 o["flatten"] = flatten
 return o
}

public external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {
 open var width: Int
 open var height: Int
 fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?
 fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String
 fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

public external interface CanvasRenderingContext2DSettings {
 var alpha: Boolean? /* = true */
 fun get(): Boolean? /* = true */
 fun set(value): Boolean? /* = true */
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): Boolean? {
 val o = js("{}")
 o["alpha"] = alpha
 return o
}

```

```

CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {\n val o =
js("{}")\n o["alpha"] = alpha\n return o\n}\n\n/**\n * Exposes the JavaScript
[CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to
Kotlin\n\n * public external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing,
CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters, CanvasRect, CanvasDrawPath, CanvasUserInterface,
CanvasText, CanvasDrawImage, CanvasHitRegion, CanvasImageData, CanvasPathDrawingStyles,
CanvasTextDrawingStyles, CanvasPath, RenderingContext {\n open val canvas:
HTMLCanvasElement\n}\n\npublic external interface CanvasState {\n fun save()\n fun restore()\n}\n\npublic
external interface CanvasTransform {\n fun scale(x: Double, y: Double)\n fun rotate(angle: Double)\n fun
translate(x: Double, y: Double)\n fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f:
Double)\n fun getTransform(): DOMMatrix\n fun setTransform(a: Double, b: Double, c: Double, d: Double, e:
Double, f: Double)\n fun setTransform(transform: dynamic = definedExternally)\n fun
resetTransform()\n}\n\npublic external interface CanvasCompositing {\n var globalAlpha: Double\n var
globalCompositeOperation: String\n}\n\npublic external interface CanvasImageSmoothing
{\n var imageSmoothingEnabled: Boolean\n var imageSmoothingQuality:
ImageSmoothingQuality\n}\n\npublic external interface CanvasFillStrokeStyles {\n var strokeStyle: dynamic\n
get() = definedExternally\n set(value) = definedExternally\n var fillStyle: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n fun createLinearGradient(x0: Double, y0: Double, x1:
Double, y1: Double): CanvasGradient\n fun createRadialGradient(x0: Double, y0: Double, r0: Double, x1:
Double, y1: Double, r1: Double): CanvasGradient\n fun createPattern(image: CanvasImageSource, repetition:
String): CanvasPattern?\n}\n\npublic external interface CanvasShadowStyles {\n var shadowOffsetX: Double\n
var shadowOffsetY: Double\n var shadowBlur: Double\n var shadowColor: String\n}\n\npublic external
interface CanvasFilters {\n var filter: String\n}\n\npublic external interface CanvasRect {\n fun clearRect(x:
Double, y: Double,
w: Double, h: Double)\n fun fillRect(x: Double, y: Double, w: Double, h: Double)\n fun strokeRect(x: Double,
y: Double, w: Double, h: Double)\n}\n\npublic external interface CanvasDrawPath {\n fun beginPath()\n fun
fill(fillRule: CanvasFillRule = definedExternally)\n fun fill(path: Path2D, fillRule: CanvasFillRule =
definedExternally)\n fun stroke()\n fun stroke(path: Path2D)\n fun clip(fillRule: CanvasFillRule =
definedExternally)\n fun clip(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n fun resetClip()\n
fun isPointInPath(x: Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n fun
isPointInPath(path: Path2D, x: Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n fun
isPointInStroke(x: Double, y: Double): Boolean\n fun isPointInStroke(path: Path2D, x: Double, y: Double):
Boolean\n}\n\npublic external interface CanvasUserInterface {\n fun drawFocusIfNeeded(element: Element)\n
fun drawFocusIfNeeded(path: Path2D, element: Element)\n fun scrollPathIntoView()\n fun
scrollPathIntoView(path: Path2D)\n}\n\npublic external interface CanvasText {\n fun fillText(text: String, x:
Double, y: Double, maxWidth: Double = definedExternally)\n fun strokeText(text: String, x: Double, y: Double,
maxWidth: Double = definedExternally)\n fun measureText(text: String): TextMetrics\n}\n\npublic external
interface CanvasDrawImage {\n fun drawImage(image: CanvasImageSource, dx: Double, dy: Double)\n fun
drawImage(image: CanvasImageSource, dx: Double, dy: Double, dw: Double, dh: Double)\n fun
drawImage(image: CanvasImageSource, sx: Double, sy: Double, sw: Double, sh: Double, dx: Double, dy: Double,
dw: Double, dh: Double)\n}\n\npublic external interface CanvasHitRegion {\n fun addHitRegion(options:
HitRegionOptions = definedExternally)\n fun removeHitRegion(id: String)\n fun clearHitRegions()\n}\n\npublic
external interface CanvasImageData {\n fun
createImageData(sw: Double, sh: Double): ImageData\n fun createImageData(imagedata: ImageData):
ImageData\n fun getImageData(sx: Double, sy: Double, sw: Double, sh: Double): ImageData\n fun
putImageData(imagedata: ImageData, dx: Double, dy: Double)\n fun putImageData(imagedata: ImageData, dx:
Double, dy: Double, dirtyX: Double, dirtyY: Double, dirtyWidth: Double, dirtyHeight: Double)\n}\n\npublic

```

```

external interface CanvasPathDrawingStyles {\n var lineWidth: Double\n var lineCap: CanvasLineCap\n var lineJoin: CanvasLineJoin\n var miterLimit: Double\n var lineDashOffset: Double\n fun setLineDash(segments: Array<Double>)\n fun getLineDash(): Array<Double>}\n\npublic external interface CanvasTextDrawingStyles {\n var font: String\n var textAlign: CanvasTextAlign\n var textBaseline: CanvasTextBaseline\n var direction: CanvasDirection}\n\npublic external interface CanvasPath {\n fun closePath()\n fun moveTo(x: Double, y: Double)\n fun lineTo(x: Double, y: Double)\n fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n fun rect(x: Double, y: Double, w: Double, h: Double)\n fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean = definedExternally)\n fun ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin\n */\n\npublic external abstract class CanvasGradient {\n fun addColorStop(offset: Double, color: String)\n}\n\n/**\n * Exposes the JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin\n */\n\npublic external abstract class CanvasPattern {\n fun setTransform(transform: dynamic = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin\n */\n\npublic external abstract class TextMetrics {\n open val width: Double\n open val actualBoundingBoxLeft: Double\n open val actualBoundingBoxRight: Double\n open val fontBoundingBoxAscent: Double\n open val fontBoundingBoxDescent: Double\n open val actualBoundingBoxAscent: Double\n open val actualBoundingBoxDescent: Double\n open val emHeightAscent: Double\n open val emHeightDescent: Double\n open val hangingBaseline: Double\n open val alphabeticBaseline: Double\n open val ideographicBaseline: Double}\n\npublic external interface HitRegionOptions {\n var path: Path2D? /* = null */\n fun get() = definedExternally\n set(value) = definedExternally\n var fillRule: CanvasFillRule? /* = CanvasFillRule.NONZERO */\n fun get() = definedExternally\n set(value) = definedExternally\n var id: String? /* = "" */\n fun get() = definedExternally\n set(value) = definedExternally\n var parentID: String? /* = null */\n fun get() = definedExternally\n set(value) = definedExternally\n var cursor: String? /* = "inherit" */\n fun get() = definedExternally\n set(value) = definedExternally\n var control: Element? /* = null */\n fun get() = definedExternally\n set(value) = definedExternally\n var label: String? /* = null */\n fun get() = definedExternally\n set(value) = definedExternally\n var role: String? /* = null */\n fun get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun HitRegionOptions(path: Path2D? = null, fillRule: CanvasFillRule? = CanvasFillRule.NONZERO, id: String? = "", parentID: String? = null, cursor: String? = "inherit", control: Element? = null, label: String? = null, role: String? = null): HitRegionOptions {\n val o = js("{}")\n o["path"] = path\n o["fillRule"] = fillRule\n o["id"] = id\n o["parentID"] = parentID\n o["cursor"] = cursor\n o["control"] = control\n o["label"] = label\n o["role"] = role\n return o\n}\n\n/**\n * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin\n */\n\npublic external open class ImageData : ImageBitmapSource, TexImageSource {\n constructor(sw: Int, sh: Int)\n constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)\n open val width: Int\n open val height: Int\n open val data: Uint8ClampedArray\n}\n\n/**\n * Exposes the JavaScript [Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin\n */\n\npublic external open class Path2D() : CanvasPath {\n constructor(path: Path2D)\n constructor(paths: Array<Path2D>, fillRule: CanvasFillRule = definedExternally)\n constructor(d: String)\n fun addPath(path: Path2D, transform: dynamic = definedExternally)\n override fun closePath()\n override fun

```

```

moveTo(x: Double, y: Double)\n override fun lineTo(x: Double, y: Double)\n override fun
quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n override fun bezierCurveTo(cp1x: Double,
cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n override fun arcTo(x1: Double, y1: Double,
x2: Double, y2: Double, radius: Double)\n override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double,
radiusX: Double, radiusY: Double, rotation: Double)\n override fun rect(x: Double, y: Double, w: Double, h:
Double)\n override fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double,
anticlockwise:
Boolean /* = definedExternally */) \n override fun ellipse(x: Double, y: Double, radiusX: Double, radiusY:
Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally
*/) \n} \n\n/** \n * Exposes the JavaScript
[ImageBitmapRenderingContext](https://developer.mozilla.org/en/docs/Web/API/ImageBitmapRenderingContext)
to Kotlin\n * \npublic external abstract class ImageBitmapRenderingContext {\n open val canvas:
HTMLCanvasElement\n fun transferFromImageBitmap(bitmap: ImageBitmap?) \n} \n\npublic external interface
ImageBitmapRenderingContextSettings {\n var alpha: Boolean? /* = true */ \n get() = definedExternally\n set(value) = definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n@kotlin.internal.InlineOnly\npublic inline fun
ImageBitmapRenderingContextSettings(alpha: Boolean? = true): ImageBitmapRenderingContextSettings {\n val o
= js("{}") \n o["alpha"] = alpha \n return o \n} \n\n/** \n
* Exposes the JavaScript
[CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin\n
* \npublic external abstract class CustomElementRegistry {\n fun define(name: String, constructor: () -> dynamic,
options: ElementDefinitionOptions = definedExternally) \n fun get(name: String): Any? \n fun
whenDefined(name: String): Promise<Unit> \n} \n\npublic external interface ElementDefinitionOptions {\n var
extends: String? \n get() = definedExternally \n set(value) =
definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n@kotlin.internal.InlineOnly\npublic inline fun ElementDefinitionOptions(extends:
String? = undefined): ElementDefinitionOptions {\n val o = js("{}") \n o["extends"] = extends \n return
o \n} \n\npublic external interface ElementContentEditable {\n var contentEditable: String \n val
isContentEditable: Boolean \n} \n\n/** \n * Exposes the JavaScript
[DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer)
to Kotlin\n * \npublic external abstract class DataTransfer {\n open var dropEffect: String \n open var
effectAllowed: String \n open val items: DataTransferItemList \n open val types: Array<out String> \n open val
files: FileList \n fun setDragImage(image: Element, x: Int, y: Int) \n fun getData(format: String): String \n fun
setData(format: String, data: String) \n fun clearData(format: String = definedExternally) \n} \n\n/** \n * Exposes
the JavaScript [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to
Kotlin\n * \npublic external abstract class DataTransferItemList {\n open val length: Int \n fun add(data: String,
type: String): DataTransferItem? \n fun add(data: File): DataTransferItem? \n fun remove(index: Int) \n fun
clear() \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n@kotlin.internal.InlineOnly\npublic inline operator fun
DataTransferItemList.get(index:
Int): DataTransferItem? = asDynamic()[index] \n\n/** \n * Exposes the JavaScript
[DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin\n * \npublic
external abstract class DataTransferItem {\n open val kind: String \n open val type: String \n fun
getAsString(_callback: ((String) -> Unit)?) \n fun getAsFile(): File? \n} \n\n/** \n * Exposes the JavaScript
[DragEvent](https://developer.mozilla.org/en/docs/Web/API/TouchEvent) to Kotlin\n * \npublic external open class
DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {\n open val
dataTransfer: DataTransfer? \n\n companion object {\n val NONE: Short \n val CAPTURING_PHASE:
Short \n val AT_TARGET: Short \n val BUBBLING_PHASE: Short \n } \n} \n\npublic external interface

```

```

DragEventInit : MouseEventInit {\n var dataTransfer: DataTransfer? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun DragEventInit(dataTransfer:
DataTransfer? = null, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0,
buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false,
modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): DragEventInit {\n val o = js("{}")\n o["dataTransfer"]
 = dataTransfer\n o["screenX"] = screenX\n o["screenY"] = screenY\n o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n o["buttons"] = buttons\n o["relatedTarget"] =
relatedTarget\n o["region"] = region\n o["ctrlKey"] = ctrlKey\n o["shiftKey"] = shiftKey\n o["altKey"]
 = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"] = modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n o["modifierFnLock"] =
modifierFnLock\n o["modifierHyper"] = modifierHyper\n o["modifierNumLock"] = modifierNumLock\n o["modifierScrollLock"] = modifierScrollLock\n o["modifierSuper"] = modifierSuper\n o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] =
view\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] =
composed\n return
o}\n}\n\n/**\n * Exposes the JavaScript [Window](https://developer.mozilla.org/en/docs/Web/API/Window) to
Kotlin\n */\npublic external abstract class Window : EventTarget, GlobalEventHandlers, WindowEventHandlers,
WindowOrWorkerGlobalScope, WindowSessionStorage, WindowLocalStorage, GlobalPerformance,
UnionMessagePortOrWindowProxy {\n open val window: Window\n open val self: Window\n open val
document: Document\n open var name: String\n open val location: Location\n open val history: History\n open val customElements: CustomElementRegistry\n open val locationbar: BarProp\n open val menubar:
BarProp\n open val personalbar: BarProp\n open val scrollbars: BarProp\n open val statusbar: BarProp\n open val toolbar: BarProp\n open var status: String\n open val closed: Boolean\n open val frames: Window\n open val length: Int\n open val top: Window\n open var opener: Any?\n open val parent: Window\n open val
frameElement: Element?\n
 open val navigator: Navigator\n open val applicationCache: ApplicationCache\n open val external: External\n open val screen: Screen\n open val innerWidth: Int\n open val innerHeight: Int\n open val scrollX: Double\n open val pageXOffset: Double\n open val scrollY: Double\n open val pageYOffset: Double\n open val
screenX: Int\n open val screenY: Int\n open val outerWidth: Int\n open val outerHeight: Int\n open val
devicePixelRatio: Double\n fun close()\n fun stop()\n fun focus()\n fun blur()\n fun open(url: String =
definedExternally, target: String = definedExternally, features: String = definedExternally): Window?\n fun
alert()\n fun alert(message: String)\n fun confirm(message: String = definedExternally): Boolean\n fun
prompt(message: String = definedExternally, default: String = definedExternally): String?\n fun print()\n fun
requestAnimationFrame(callback: (Double) -> Unit): Int\n fun cancelAnimationFrame(handle:
Int)\n fun postMessage(message: Any?, targetOrigin: String, transfer: Array<dynamic> = definedExternally)\n fun
captureEvents()\n fun releaseEvents()\n fun matchMedia(query: String): MediaQueryList\n fun moveTo(x:
Int, y: Int)\n fun moveBy(x: Int, y: Int)\n fun resizeTo(x: Int, y: Int)\n fun resizeBy(x: Int, y: Int)\n fun
scroll(options: ScrollToOptions = definedExternally)\n fun scroll(x: Double, y: Double)\n fun scrollTo(options:
ScrollToOptions = definedExternally)\n fun scrollTo(x: Double, y: Double)\n fun scrollBy(options:
ScrollToOptions = definedExternally)\n fun scrollBy(x: Double, y: Double)\n fun getComputedStyle(elt:
Element, pseudoElt: String? = definedExternally):
CSSStyleDeclaration}\n}\n\n@Suppress("INVISIBLE_REFERENCE",

```

```

\@kotlin.internal.InlineOnly\npublic inline operator fun Window.get(name: String):
dynamic = asDynamic()[name]\n\npublic external abstract class BarProp {\n open val visible:
Boolean\n}\n\n/**\n * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to
Kotlin\n *\npublic external abstract class History {\n open val length: Int\n open var scrollRestoration:
ScrollRestoration\n open val state: Any?\n fun go(delta: Int = definedExternally)\n fun back()\n fun
forward()\n fun pushState(data: Any?, title: String, url: String? = definedExternally)\n fun replaceState(data:
Any?, title: String, url: String? = definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin\n *\npublic external abstract class
Location {\n open var href: String\n open val origin: String\n open var protocol: String\n open var host:
String\n open var hostname: String\n open var port: String\n open var pathname: String\n open var search:
String\n open var hash: String\n open val ancestorOrigins: Array<out String>\n fun assign(url:
String)\n fun replace(url: String)\n fun reload()\n}\n\n/**\n * Exposes the JavaScript
[PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin\n *\npublic external
open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {\n open val
state: Any?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface PopStateEventInit
: EventInit {\n var state: Any? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PopStateEventInit(state: Any? = null,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PopStateEventInit {\n val o
= js(\\"({})\")\n o[\"state\"] = state\n o[\"bubbles\"] = bubbles\n
 o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin\n *\npublic
external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) :
Event {\n open val oldURL: String\n open val newURL: String\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface HashChangeEventInit : EventInit {\n var oldURL: String? /* = \\"\" */\n
get() = definedExternally\n set(value) = definedExternally\n var newURL: String? /* = \\"\" */\n
get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun HashChangeEventInit(oldURL:
String?
= \\"\", newURL: String? = \\"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): HashChangeEventInit {\n val o = js(\\"({})\")\n o[\"oldURL\"] = oldURL\n o[\"newURL\"] =
newURL\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript
[PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin\n *\npublic
external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) :
Event {\n open val persisted: Boolean\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface PageTransitionEventInit : EventInit {\n var persisted: Boolean? /* = false */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PageTransitionEventInit(persisted:
Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
PageTransitionEventInit {\n val o = js(\\"({})\")\n o[\"persisted\"] = persisted\n o[\"bubbles\"] = bubbles\n
 o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[BeforeUnloadEvent](https://developer.mozilla.org/en/docs/Web/API/BeforeUnloadEvent) to Kotlin\n *\npublic
external open class BeforeUnloadEvent : Event {\n var returnValue: String\n\n companion object {\n val
NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val

```





```

definedExternally\n var onclick: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onclose: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var oncontextmenu: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var oncuechange:
((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondblclick:
((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondrag:
((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragend:
((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ondragenter: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var ondragexit: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var ondragleave: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var ondragover: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var ondragstart: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondrop: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondurationchange: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onemptied: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onended: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onerror: ((dynamic, String, Int, Int, Any?) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onfocus: ((FocusEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oninput: ((InputEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oninvalid: ((Event) ->
dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onkeydown: ((KeyboardEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onkeypress: ((KeyboardEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onkeyup: ((KeyboardEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onload: ((Event) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n var onloadeddata: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onloadedmetadata: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onloadend: ((Event) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n var onloadstart: ((ProgressEvent) -> dynamic)?\n
get() = definedExternally\n set(value)
= definedExternally\n var onmousedown: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onmouseenter: ((MouseEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onmouseleave: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onmousemove: ((MouseEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseout:
((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onmouseover: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onmouseup: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onwheel: ((WheelEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n
var onpause: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onplay: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onplaying: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onprogress: ((ProgressEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onratechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onreset: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onresize: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var

```

```

onscroll: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onseeked: ((Event) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n var onseeking: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onselect: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onshow: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onstalled: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onsubmit: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onsuspend: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var ontimeupdate: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var ontoggle: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n
var onvolumechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onwaiting: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var ongotpointercapture: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onlostpointercapture: ((PointerEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onpointerdown: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onpointermove: ((PointerEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onpointerup:
((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onpointercancel: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n
var onpointerover: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onpointerout: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onpointerenter: ((PointerEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onpointerleave: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n**\n * Exposes the JavaScript
[WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEventHandlers) to Kotlin\n
*\n\npublic external interface WindowEventHandlers {\n var onafterprint: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onbeforeprint: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onbeforeunload: ((BeforeUnloadEvent)
-> String)?\n get() = definedExternally\n set(value) = definedExternally\n var onhashchange:
((HashChangeEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onlanguagechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onmessage: ((MessageEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onoffline: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var ononline: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onpagehide: ((PageTransitionEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onpageshow: ((PageTransitionEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onpopstate: ((PopStateEvent)
-> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onrejectionhandled:
((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onstorage:
((StorageEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onunload: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\npublic external interface DocumentAndElementEventHandlers {\n var oncopy:
((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncut:
((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onpaste: ((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n

```

```

set(value) = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[WindowOrWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WindowOrWorkerGlobalScope)
to Kotlin\n */\npublic external interface WindowOrWorkerGlobalScope {\n val origin: String\n val caches:
CacheStorage\n fun btoa(data: String): String\n fun atob(data: String): String\n fun setTimeout(handler:
dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearTimeout(handle: Int =
definedExternally)\n fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?):
Int\n fun clearInterval(handle: Int = definedExternally)\n fun createImageBitmap(image: ImageBitmapSource,
options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun createImageBitmap(image:
ImageBitmapSource, sx: Int, sy: Int, sw: Int, sh: Int, options: ImageBitmapOptions = definedExternally):
Promise<ImageBitmap>\n fun fetch(input: dynamic,
init: RequestInit = definedExternally): Promise<Response>\n}\n\n/**\n * Exposes the JavaScript
[Navigator](https://developer.mozilla.org/en/docs/Web/API/Navigator) to Kotlin\n */\npublic external abstract class
Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies,
NavigatorPlugins, NavigatorConcurrentHardware {\n open val clipboard: Clipboard\n open val mediaDevices:
MediaDevices\n open val maxTouchPoints: Int\n open val serviceWorker: ServiceWorkerContainer\n fun
requestMediaKeySystemAccess(keySystem: String, supportedConfigurations:
Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>\n fun getUserMedia(constraints:
MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)\n fun
vibrate(pattern: dynamic): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorID](https://developer.mozilla.org/en/docs/Web/API/NavigatorID) to Kotlin\n */\npublic external interface
NavigatorID {\n val appCodeName: String\n val appName: String\n val appVersion: String\n val platform:
String\n val product: String\n val productSub: String\n val userAgent: String\n val vendor: String\n val
vendorSub: String\n val oscpu: String\n fun taintEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorLanguage](https://developer.mozilla.org/en/docs/Web/API/NavigatorLanguage) to Kotlin\n */\npublic
external interface NavigatorLanguage {\n val language: String\n val languages: Array<out String>\n}\n\npublic
external interface NavigatorContentUtils {\n fun registerProtocolHandler(scheme: String, url: String, title:
String)\n fun registerContentHandler(mimeType: String, url: String, title: String)\n fun
isProtocolHandlerRegistered(scheme: String, url: String): String\n fun isContentHandlerRegistered(mimeType:
String, url: String): String\n fun unregisterProtocolHandler(scheme: String, url: String)\n fun
unregisterContentHandler(mimeType:
String, url: String)\n}\n\npublic external interface NavigatorCookies {\n val cookieEnabled: Boolean\n}\n\n/**\n
* Exposes the JavaScript [NavigatorPlugins](https://developer.mozilla.org/en/docs/Web/API/NavigatorPlugins) to
Kotlin\n */\npublic external interface NavigatorPlugins {\n val plugins: PluginArray\n val mimeTypes:
MimeTypeArray\n fun javaEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[PluginArray](https://developer.mozilla.org/en/docs/Web/API/PluginArray) to Kotlin\n */\npublic external abstract
class PluginArray : ItemArrayLike<Plugin> {\n fun refresh(reload: Boolean = definedExternally)\n override fun
item(index: Int): Plugin?\n fun namedItem(name: String):
Plugin?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(index: Int):
Plugin? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline operator fun PluginArray.get(name: String): Plugin? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MimeTypeArray](https://developer.mozilla.org/en/docs/Web/API/MimeTypeArray) to Kotlin\n */\npublic external
abstract class MimeTypeArray : ItemArrayLike<MimeType> {\n override fun item(index: Int): MimeType?\n fun
namedItem(name: String): MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(index:
Int): MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(name:

```

```

String): MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin\n *\npublic external abstract class Plugin
: ItemArrayLike<MimeType> {\n open val name: String\n open val description: String\n open
 val filename: String\n override fun item(index: Int): MimeType?\n fun namedItem(name: String):
MimeType?\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(index: Int):
MimeType? = asDynamic()[index]\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(name: String):
MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin\n *\npublic external abstract
class MimeType {\n open val type: String\n open val description: String\n open val suffixes: String\n open
val enabledPlugin: Plugin\n}\n\n/**\n * Exposes the JavaScript
[ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin\n *\npublic external
abstract class ImageBitmap : CanvasImageSource, TexImageSource {\n open val width: Int\n open val height:
Int\n fun close()\n}\n\npublic external interface ImageBitmapOptions {\n var imageOrientation:
ImageOrientation? /* = ImageOrientation.NONE */\n get() = definedExternally\n set(value) =
definedExternally\n var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */\n get() =
definedExternally\n set(value) = definedExternally\n var colorSpaceConversion: ColorSpaceConversion? /* =
ColorSpaceConversion.DEFAULT */\n get() = definedExternally\n set(value) = definedExternally\n var
resizeWidth: Int?\n get() = definedExternally\n set(value) = definedExternally\n var resizeHeight: Int?\n
 get() = definedExternally\n set(value) = definedExternally\n var resizeQuality: ResizeQuality? /* =
ResizeQuality.LOW */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n@kotlin.internal.InlineOnly\npublic inline fun
ImageBitmapOptions(imageOrientation:
ImageOrientation? = ImageOrientation.NONE, premultiplyAlpha: PremultiplyAlpha? =
PremultiplyAlpha.DEFAULT, colorSpaceConversion: ColorSpaceConversion? =
ColorSpaceConversion.DEFAULT, resizeWidth: Int? = undefined, resizeHeight: Int? = undefined, resizeQuality:
ResizeQuality? = ResizeQuality.LOW): ImageBitmapOptions {\n val o = js(\\"({})\\")\n o[\"imageOrientation\"]
= imageOrientation\n o[\"premultiplyAlpha\"] = premultiplyAlpha\n o[\"colorSpaceConversion\"] =
colorSpaceConversion\n o[\"resizeWidth\"] = resizeWidth\n o[\"resizeHeight\"] = resizeHeight\n
o[\"resizeQuality\"] = resizeQuality\n return o\n}\n\n/**\n * Exposes the JavaScript
[MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin\n *\npublic external open
class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {\n open val data:
Any?\n open val origin: String\n open val lastEventId: String\n
 open val source: UnionMessagePortOrWindowProxy?\n open val ports: Array<out MessagePort>\n fun
initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId:
String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)\n\n companion object {\n
val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MessageEventInit : EventInit {\n var data: Any?
/* = null */\n get() = definedExternally\n set(value) = definedExternally\n var origin: String? /* = \\"\" */\n
get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String? /* = \\"\" */\n
get() = definedExternally\n set(value) = definedExternally\n var source: UnionMessagePortOrWindowProxy?
/* = null */\n get() = definedExternally\n set(value) = definedExternally\n var ports:
Array<MessagePort>? /* = arrayOf() */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n@kotlin.internal.InlineOnly\npublic inline fun MessageEventInit(data: Any? = null,
origin: String? = \\"\", lastEventId: String? = \\"\", source: UnionMessagePortOrWindowProxy? = null, ports:

```

```

Array<MessagePort>? = arrayOf(), bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): MessageEventInit {\n val o = js("{}")\n o["data"] = data\n o["origin"] = origin\n
o["lastEventId"] = lastEventId\n o["source"] = source\n o["ports"] = ports\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin\n *\npublic external open
class EventSource(url: String, eventSourceInitDict: EventSourceInit
= definedExternally) : EventTarget {\n open val url: String\n open val withCredentials: Boolean\n open val
readyState: Short\n var onopen: ((Event) -> dynamic)?\n var onmessage: ((MessageEvent) -> dynamic)?\n var
onerror: ((Event) -> dynamic)?\n fun close()\n\n companion object {\n val CONNECTING: Short\n val
OPEN: Short\n val CLOSED: Short\n }\n}\n\npublic external interface EventSourceInit {\n var
withCredentials: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventSourceInit(withCredentials:
Boolean? = false): EventSourceInit {\n val o = js("{}")\n o["withCredentials"] = withCredentials\n return
o\n}\n\n/**\n * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to
Kotlin\n *\npublic external open class WebSocket(url:
String, protocols: dynamic = definedExternally) : EventTarget {\n open val url: String\n open val readyState:
Short\n open val bufferedAmount: Number\n var onopen: ((Event) -> dynamic)?\n var onerror: ((Event) ->
dynamic)?\n var onclose: ((Event) -> dynamic)?\n open val extensions: String\n open val protocol: String\n
var onmessage: ((MessageEvent) -> dynamic)?\n var binaryType: BinaryType\n fun close(code: Short =
definedExternally, reason: String = definedExternally)\n fun send(data: String)\n fun send(data: Blob)\n fun
send(data: ArrayBuffer)\n fun send(data: ArrayBufferView)\n\n companion object {\n val CONNECTING:
Short\n val OPEN: Short\n val CLOSING: Short\n val CLOSED: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin\n *\npublic external
open class CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally)
: Event {\n open val wasClean: Boolean\n open val code: Short\n open val reason: String\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface CloseEventInit : EventInit {\n var wasClean:
Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var code: Short? /*
= 0 */\n get() = definedExternally\n set(value) = definedExternally\n var reason: String? /* = "" */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CloseEventInit(wasClean: Boolean? =
false, code: Short? = 0, reason: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CloseEventInit {\n val o = js("{}")\n o["wasClean"]
= wasClean\n o["code"] = code\n o["reason"] = reason\n o["bubbles"] = bubbles\n o["cancelable"] =
cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin\n *\npublic external
open class MessageChannel {\n open val port1: MessagePort\n open val port2: MessagePort\n}\n\n/**\n *
Exposes the JavaScript [MessagePort](https://developer.mozilla.org/en/docs/Web/API/MessagePort) to Kotlin\n
*\npublic external abstract class MessagePort : EventTarget, UnionMessagePortOrWindowProxy,
UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n open var onmessage:
((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?, transfer: Array<dynamic> =
definedExternally)\n fun start()\n fun close()\n}\n\n/**\n * Exposes the JavaScript
[BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel)
to Kotlin\n *\npublic external open class BroadcastChannel(name: String) : EventTarget {\n open val name:
String\n var onmessage: ((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?)\n fun
close()\n}\n\n/**\n * Exposes the JavaScript
[WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin\n *\npublic

```

```

external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope, GlobalPerformance {
 open val self: WorkerGlobalScope
 open val location: WorkerLocation
 open val navigator: WorkerNavigator
 open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
 open var onlanguagechange: ((Event) -> dynamic)?
 open var onoffline: ((Event) -> dynamic)?
 open var ononline: ((Event) -> dynamic)?
 open var onrejectionhandled: ((Event) -> dynamic)?
 open var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?
 fun importScripts(vararg urls: String)
}
Exposes the JavaScript

[DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to Kotlin
public external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {
 open var onmessage: ((MessageEvent) -> dynamic)?
 fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
 fun close()
}
Exposes the JavaScript

[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to Kotlin
public external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {
 open val name: String
 open val applicationCache: ApplicationCache
 open var onconnect: ((Event) -> dynamic)?
 fun close()
}
Exposes the JavaScript

[AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin
public external interface AbstractWorker {
 var onerror: ((Event) -> dynamic)?
 get() = definedExternally
 set(value) = definedExternally
}
Exposes the JavaScript

[Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin
public external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
 var onmessage: ((MessageEvent) -> dynamic)?
 override var onerror: ((Event) -> dynamic)?
 fun terminate()
 fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}
public external interface WorkerOptions {
 var type: WorkerType? /* = WorkerType.CLASSIC */
 get() = definedExternally
 set(value) = definedExternally
 var credentials: RequestCredentials? /* = RequestCredentials.OMIT */
 get() = definedExternally
 set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun WorkerOptions(type: WorkerType? = WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {
 val o = js("{}")
 o["type"] = type
 o["credentials"] = credentials
 return o
}
Exposes the JavaScript

[SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin
public external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
 open val port: MessagePort
 override var onerror: ((Event) -> dynamic)?
}
Exposes the JavaScript

[NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin
public external interface NavigatorConcurrentHardware {
 val hardwareConcurrency: Number
}
Exposes the JavaScript

[WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin
public external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {
 open val serviceWorker: ServiceWorkerContainer
}
Exposes the JavaScript

[WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin
public external abstract class WorkerLocation {
 open val href: String
 open val origin: String
 open val protocol: String
 open val host: String
 open val hostname: String
 open val port: String
 open val pathname: String
 open val search: String
 open val hash: String
}
Exposes the JavaScript

[Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin
public external abstract class Storage {
 open val length: Int
 fun key(index: Int): String?
 fun removeItem(key: String)
 fun clear()
 fun getItem(key: String): String?
 fun setItem(key: String, value: String)
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun

```

```

Storage.get(key: String): String? = asDynamic()[key]\n\n@Suppress("\n\nINVISIBLE_REFERENCE",
"\n\nINVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun Storage.set(key: String, value:
String) { asDynamic()[key] = value }\n\n/**\n * Exposes the JavaScript
[WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin\n
*\n\npublic external interface WindowSessionStorage {\n val sessionStorage: Storage\n}\n\n/**\n * Exposes the
JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to
Kotlin\n
*\n\npublic external interface WindowLocalStorage {\n val localStorage: Storage\n}\n\n/**\n * Exposes
the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin\n
*\n\npublic
external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {\n
open val key: String?\n open val oldValue: String?\n open val newValue:
String?\n open val url: String\n open val storageArea: Storage?\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface StorageEventInit : EventInit {\n var key: String? /* = null */\n get()
= definedExternally\n set(value) = definedExternally\n var oldValue: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var newValue: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var url: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var storageArea: Storage? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("\n\nINVISIBLE_REFERENCE",
"\n\nINVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun StorageEventInit(key:
String? = null, oldValue: String? = null, newValue: String? = null, url: String? = "", storageArea: Storage? = null,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): StorageEventInit {\n val o
= js("\n\n({})")\n o["key"] = key\n o["oldValue"] = oldValue\n o["newValue"] = newValue\n o["url"] =
url\n o["storageArea"] = storageArea\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n
o["composed"] = composed\n return o\n}\n\npublic external abstract class HTMLAppletElement :
HTMLElement {\n open var align: String\n open var alt: String\n open var archive: String\n open var code:
String\n open var codeBase: String\n open var height: String\n open var hspace: Int\n open var name:
String\n open var _object: String\n open var vspace: Int\n open var width: String\n\n companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin\n
*\n\npublic external abstract class HTMLMarqueeElement : HTMLElement {\n open var behavior: String\n open
var bgColor: String\n open var direction: String\n open var height: String\n
 open var hspace: Int\n open var loop: Int\n open var scrollAmount: Int\n open var scrollDelay: Int\n open
var trueSpeed: Boolean\n open var vspace: Int\n open var width: String\n open var onbounce: ((Event) ->
dynamic)?\n open var onfinish: ((Event) -> dynamic)?\n open var onstart: ((Event) -> dynamic)?\n fun
start()\n fun stop()\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

```



```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin\n
*\npublic external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {\n open var
cols: String\n open var rows: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
HTMLFrameElement : HTMLElement {\n open var name: String\n open var scrolling: String\n open var src:
String\n open var frameBorder: String\n open var longDesc: String\n open var noResize: Boolean\n open val
contentDocument: Document?\n open val contentWindow: Window?\n open var marginHeight: String\n open
var marginWidth: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
HTMLDirectoryElement : HTMLElement {\n open var compact: Boolean\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFontElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFontElement) to Kotlin\n
*\npublic external abstract class HTMLFontElement : HTMLElement {\n open var color: String\n open var face: String\n
open var size: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n

```

```

 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n \n \n public external
interface External { \n fun AddSearchProvider() \n fun IsSearchProviderInstalled() \n } \n \n public external interface
EventInit { \n var bubbles: Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n var cancelable: Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n var composed: Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun EventInit(bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): EventInit { \n val o = js("{}") \n o["bubbles"] =
bubbles \n o["cancelable"] = cancelable \n o["composed"] = composed \n return o \n } \n \n /** \n * Exposes the
JavaScript [CustomEvent](https://developer.mozilla.org/en/docs/Web/API/CustomEvent) to Kotlin \n * \n public
external open class CustomEvent(type: String, eventInitDict: CustomEventInit = definedExternally) : Event { \n
open val detail: Any? \n fun initCustomEvent(type: String, bubbles: Boolean, cancelable: Boolean, detail:
Any?) \n \n companion object { \n val NONE: Short \n val CAPTURING_PHASE: Short \n val
AT_TARGET: Short \n val BUBBLING_PHASE: Short \n } \n \n public external interface CustomEventInit :
EventInit { \n var detail: Any? /* = null */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public
inline fun CustomEventInit(detail: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CustomEventInit { \n val o = js("{}") \n o["detail"] = detail \n o["bubbles"] = bubbles \n
o["cancelable"] = cancelable \n o["composed"] = composed \n return o \n } \n \n public external interface
EventListenerOptions { \n var capture: Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun EventListenerOptions(capture:
Boolean? = false): EventListenerOptions { \n val o = js("{}") \n o["capture"] = capture \n return
o \n } \n \n public external interface AddEventListenerOptions : EventListenerOptions { \n var passive: Boolean? /* =
false */ \n get() = definedExternally \n set(value) = definedExternally \n var once: Boolean? /* = false */ \n
get() = definedExternally \n
 set(value) = definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun AddEventListenerOptions(passive:
Boolean? = false, once: Boolean? = false, capture: Boolean? = false): AddEventListenerOptions { \n val o =
js("{}") \n o["passive"] = passive \n o["once"] = once \n o["capture"] = capture \n return o \n } \n \n public
external interface NonElementParentNode { \n fun getElementById(elementId: String): Element? \n } \n \n /** \n *
Exposes the JavaScript
[DocumentOrShadowRoot](https://developer.mozilla.org/en/docs/Web/API/DocumentOrShadowRoot) to Kotlin \n * \n public
external interface DocumentOrShadowRoot { \n val fullscreenElement: Element? \n get() =
definedExternally \n } \n \n /** \n * Exposes the JavaScript
[ParentNode](https://developer.mozilla.org/en/docs/Web/API/ParentNode) to Kotlin \n * \n public external interface
ParentNode { \n val children: HTMLCollection \n val firstElementChild:
Element? \n get() = definedExternally \n val lastElementChild: Element? \n get() = definedExternally \n
val childElementCount: Int \n fun prepend(vararg nodes: dynamic) \n fun append(vararg nodes: dynamic) \n fun
querySelector(selectors: String): Element? \n fun querySelectorAll(selectors: String): NodeList \n } \n \n /** \n *
Exposes the JavaScript
[NonDocumentTypeChildNode](https://developer.mozilla.org/en/docs/Web/API/NonDocumentTypeChildNode) to
Kotlin \n * \n public external interface NonDocumentTypeChildNode { \n val previousElementSibling: Element? \n
get() = definedExternally \n val nextElementSibling: Element? \n get() = definedExternally \n } \n \n /** \n *

```

```

Exposes the JavaScript [ChildNode](https://developer.mozilla.org/en/docs/Web/API/ChildNode) to Kotlin\n
*\npublic external interface ChildNode {\n fun before(vararg nodes: dynamic)\n fun after(vararg nodes:
dynamic)\n fun replaceWith(vararg nodes: dynamic)\n fun remove()\n}\n\n**\n
* Exposes the JavaScript [Slotable](https://developer.mozilla.org/en/docs/Web/API/Slotable) to Kotlin\n *\npublic
external interface Slotable {\n val assignedSlot: HTMLSlotElement?\n get() = definedExternally\n}\n\n**\n
* Exposes the JavaScript [NodeList](https://developer.mozilla.org/en/docs/Web/API/NodeList) to Kotlin\n *\npublic
external abstract class NodeList : ItemArrayLike<Node> {\n override fun item(index: Int):
Node?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun NodeList.get(index: Int):
Node? = asDynamic()[index]\n\n**\n
* Exposes the JavaScript
[HTMLCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLCollection) to Kotlin\n *\npublic
external abstract class HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {\n
override fun item(index: Int): Element?\n fun namedItem(name: String):
Element?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic
inline operator fun HTMLCollection.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(name:
String): Element? = asDynamic()[name]\n\n**\n
* Exposes the JavaScript
[MutationObserver](https://developer.mozilla.org/en/docs/Web/API/MutationObserver) to Kotlin\n *\npublic
external open class MutationObserver(callback: (Array<MutationRecord>, MutationObserver) -> Unit) {\n fun
observe(target: Node, options: MutationObserverInit = definedExternally)\n fun disconnect()\n fun
takeRecords(): Array<MutationRecord>\n}\n\n**\n
* Exposes the JavaScript
[MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin\n
\npublic external interface MutationObserverInit {\n var childList: Boolean? / = false */\n get() =
definedExternally\n set(value) = definedExternally\n
var attributes: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var
characterData: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var subtree:
Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var
attributeOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var
characterDataOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var
attributeFilter: Array<String>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MutationObserverInit(childList:
Boolean? = false, attributes: Boolean? = undefined, characterData: Boolean? = undefined, subtree: Boolean? = false,
attributeOldValue: Boolean? = undefined, characterDataOldValue:
Boolean? = undefined, attributeFilter: Array<String>? = undefined): MutationObserverInit {\n val o =
js("{}")\n o["childList"] = childList\n o["attributes"] = attributes\n o["characterData"] =
characterData\n o["subtree"] = subtree\n o["attributeOldValue"] = attributeOldValue\n
o["characterDataOldValue"] = characterDataOldValue\n o["attributeFilter"] = attributeFilter\n return
o\n}\n\n**\n
* Exposes the JavaScript
[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin\n *\npublic external
abstract class MutationRecord {\n open val type: String\n open val target: Node\n open val addedNodes:
NodeList\n open val removedNodes: NodeList\n open val previousSibling: Node?\n open val nextSibling:
Node?\n open val attributeName: String?\n open val attributeNamespace: String?\n open val oldValue:
String?\n}\n\n**\n
* Exposes the JavaScript [Node](https://developer.mozilla.org/en/docs/Web/API/Node)
to Kotlin\n *\npublic external abstract class Node : EventTarget {\n open val nodeType: Short\n open val
nodeName: String\n open val baseURI: String\n open val isConnected: Boolean\n open val ownerDocument:

```

```

Document?
 open val parentNode: Node?
 open val parentElement: Element?
 open val childNodes:
 NodeList
 open val firstChild: Node?
 open val lastChild: Node?
 open val previousSibling: Node?
 open val nextSibling: Node?
 open var nodeValue: String?
 open var textContent: String?
 fun
 getRootNode(options: GetRootNodeOptions = definedExternally): Node
 fun hasChildNodes(): Boolean
 fun
 normalize()
 fun cloneNode(deep: Boolean = definedExternally): Node
 fun isEqualNode(otherNode: Node?):
 Boolean
 fun isSameNode(otherNode: Node?): Boolean
 fun compareDocumentPosition(other: Node): Short
 fun contains(other: Node?): Boolean
 fun lookupPrefix(namespace: String?): String?
 fun
 lookupNamespaceURI(prefix: String?): String?
 fun isDefaultNamespace(namespace: String?): Boolean
 fun
 insertBefore(node: Node, child: Node?): Node
 fun appendChild(node: Node): Node
 fun replaceChild(node:
 Node, child: Node): Node
 fun removeChild(child: Node): Node
 companion object {
 val
 ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val
 CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE:
 Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val
 DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val
 DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val
 DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
 public external interface
 GetRootNodeOptions {
 var composed: Boolean? /* = false */
 get() = definedExternally
 set(value) =
 definedExternally
 }
 @Suppress("INVISIBLE_REFERENCE",
 "INVISIBLE_MEMBER")
 @kotlin.internal.InlineOnly
 public inline fun GetRootNodeOptions(composed:
 Boolean? = false): GetRootNodeOptions {
 val o = js("{}")
 o["composed"] = composed
 return
 o
 }
 /**
 * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to
 * Kotlin
 */
 public external open class Document : Node, GlobalEventHandlers,
 DocumentAndElementEventHandlers, NonElementParentNode, DocumentOrShadowRoot, ParentNode,
 GeometryUtils {
 open val implementation: DOMImplementation
 open val URL: String
 open val
 documentURI: String
 open val origin: String
 open val compatMode: String
 open val characterSet:
 String
 open val
 charset: String
 open val inputEncoding: String
 open val contentType: String
 open val doctype:
 DocumentType?
 open val documentElement: Element?
 open val location: Location?
 var domain:
 String
 open val referrer: String
 var cookie: String
 open val lastModified: String
 open val readyState:
 DocumentReadyState
 var title: String
 var dir: String
 var body: HTMLElement?
 open val head:
 HTMLHeadElement?
 open val images: HTMLCollection
 open val embeds: HTMLCollection
 open val
 plugins: HTMLCollection
 open val links: HTMLCollection
 open val forms: HTMLCollection
 open val
 scripts: HTMLCollection
 open val currentScript: HTMLScriptElement?
 open val defaultView:
 Window?
 open val activeElement: Element?
 var designMode: String
 var onreadystatechange: ((Event) ->
 dynamic)?
 var fgColor: String
 var linkColor: String
 var vlinkColor: String
 var alinkColor: String
 var bgColor: String
 open val anchors: HTMLCollection
 open val applets: HTMLCollection
 open val
 all: HTMLAllCollection
 open val scrollingElement: Element?
 open val styleSheets: StyleSheetList
 open val rootElement: SVGSVGElement?
 open val fullscreenEnabled: Boolean
 open val fullscreen: Boolean
 var onfullscreenchange: ((Event) -> dynamic)?
 var onfullscreenerror: ((Event) -> dynamic)?
 override var
 onabort: ((Event) -> dynamic)?
 override var onblur: ((FocusEvent) -> dynamic)?
 override var oncancel:
 ((Event) -> dynamic)?
 override var oncanplay: ((Event) -> dynamic)?
 override var oncanplaythrough:
 ((Event) -> dynamic)?
 override var onchange: ((Event) -> dynamic)?
 override var onclick: ((MouseEvent) ->
 dynamic)?
 override var onclose: ((Event) -> dynamic)?
 override var oncontextmenu: ((MouseEvent) ->
 dynamic)?
 override var oncuechange: ((Event) -> dynamic)?
 override var ondblclick: ((MouseEvent)

```

```

-> dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) ->
dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var ondragexit: ((DragEvent) ->
dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover: ((DragEvent) ->
dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) ->
dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) ->
dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) -
> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup:
((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata:
((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend:
((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown:
((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var
onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n
override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) ->
dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) ->
dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange:
((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var onresize: ((Event) ->
dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked: ((Event) -> dynamic)?\n
override var onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) -> dynamic)?\n override var
onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n override var onsubmit:
((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var ontimeupdate: ((Event) -
> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var onvolumechange: ((Event) ->
dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var ongotpointercapture: ((PointerEvent)
-> dynamic)?\n override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n override var onpointerdown:
((PointerEvent) -> dynamic)?\n override var onpointermove: ((PointerEvent) -> dynamic)?\n
 override var onpointerup: ((PointerEvent) -> dynamic)?\n override var onpointercancel: ((PointerEvent) ->
dynamic)?\n override var onpointerover: ((PointerEvent) -> dynamic)?\n override var onpointerout:
((PointerEvent) -> dynamic)?\n override var onpointerenter: ((PointerEvent) -> dynamic)?\n override var
onpointerleave: ((PointerEvent) -> dynamic)?\n override var oncopy: ((ClipboardEvent) -> dynamic)?\n override
var oncut: ((ClipboardEvent) -> dynamic)?\n override var onpaste: ((ClipboardEvent) -> dynamic)?\n override
val fullscreenElement: Element?\n override val children: HTMLCollection\n override val firstElementChild:
Element?\n override val lastElementChild: Element?\n override val childElementCount: Int\n fun
getElementsByTagName(qualifiedName: String): HTMLCollection\n fun
getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n fun
getElementsByClassName(classNames: String): HTMLCollection\n fun createElement(localName:
String, options: ElementCreationOptions = definedExternally): Element\n fun createElementNS(namespace:
String?, qualifiedName: String, options: ElementCreationOptions = definedExternally): Element\n fun
createDocumentFragment(): DocumentFragment\n fun createTextNode(data: String): Text\n fun
createCDATASection(data: String): CDATASection\n fun createComment(data: String): Comment\n fun
createProcessingInstruction(target: String, data: String): ProcessingInstruction\n fun importNode(node: Node,
deep: Boolean = definedExternally): Node\n fun adoptNode(node: Node): Node\n fun
createAttribute(localName: String): Attr\n fun createAttributeNS(namespace: String?, qualifiedName: String):
Attr\n fun createEvent(`interface`: String): Event\n fun createRange(): Range\n fun createNodeIterator(root:
Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): NodeIterator\n fun
createNodeIterator(root: Node, whatToShow:

```

```

Int = definedExternally, filter: ((Node) -> Short)? = definedExternally): NodeIterator\n fun
createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally):
TreeWalker\n fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)?
= definedExternally): TreeWalker\n fun getElementByName(elementName: String): NodeList\n fun open(type:
String = definedExternally, replace: String = definedExternally): Document\n fun open(url: String, name: String,
features: String): Window\n fun close()\n fun write(vararg text: String)\n fun writeln(vararg text: String)\n
fun hasFocus(): Boolean\n fun execCommand(commandId: String, showUI: Boolean = definedExternally, value:
String = definedExternally): Boolean\n fun queryCommandEnabled(commandId: String): Boolean\n fun
queryCommandIndeterm(commandId: String): Boolean\n fun queryCommandState(commandId: String):
Boolean\n fun queryCommandSupported(commandId:
String): Boolean\n fun queryCommandValue(commandId: String): String\n fun clear()\n fun
captureEvents()\n fun releaseEvents()\n fun elementFromPoint(x: Double, y: Double): Element?\n fun
elementsFromPoint(x: Double, y: Double): Array<Element>\n fun caretPositionFromPoint(x: Double, y: Double):
CaretPosition?\n fun createTouch(view: Window, target: EventTarget, identifier: Int, pageX: Int, pageY: Int,
screenX: Int, screenY: Int): Touch\n fun createTouchList(vararg touches: Touch): TouchList\n fun
exitFullscreen(): Promise<Unit>\n override fun getElementById(elementId: String): Element?\n override fun
prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun
querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override
fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Document.get(name: String):
dynamic = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin\n */\npublic external
open class XMLDocument : Document {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n
val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface

```

```

ElementCreationOptions {\n var `is`: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun ElementCreationOptions(`is`: String?
= undefined): ElementCreationOptions {\n val o = js(\"({})\")\n o[\"is\"] = `is`\n return o\n}\n\n/**\n *
Exposes the JavaScript
[DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin\n
*\n\npublic external abstract class DOMImplementation {\n fun createDocumentType(qualifiedName: String,
publicId: String, systemId: String):
DocumentType\n fun createDocument(namespace: String?, qualifiedName: String, doctype: DocumentType? =
definedExternally): XMLDocument\n fun createHTMLDocument(title: String = definedExternally): Document\n
fun hasFeature(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin\n
*\n\npublic external
abstract class DocumentType : Node, ChildNode {\n open val name: String\n open val publicId: String\n open
val systemId: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin\n
*\n\npublic
external open class DocumentFragment : Node, NonElementParentNode, ParentNode {\n override val children:
HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild: Element?\n
override val childElementCount: Int\n override fun getElementById(elementId: String): Element?\n override fun
prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun
querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n\n
companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin\n
*\n\npublic external open
class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {\n open val mode: ShadowRootMode\n open val
host: Element\n override val fullscreenElement: Element?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n

```

```

 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin \n * \n public external abstract
class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils,
UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent,
UnionElementOrProcessingInstruction { \n open val namespaceURI: String? \n open val prefix: String? \n open
val localName: String \n open val tagName: String \n open var id: String \n open var className: String \n open
val classList: DOMTokenList \n open var slot: String \n open val attributes: NamedNodeMap \n open val
shadowRoot: ShadowRoot? \n open var scrollTop: Double \n open var scrollLeft: Double \n open val
scrollWidth: Int \n open val scrollHeight: Int \n open val clientTop: Int \n open val clientLeft: Int \n open val
clientWidth: Int \n open val clientHeight: Int \n open var innerHTML: String \n open var outerHTML: String \n
fun hasAttributes(): Boolean \n fun getAttributeNames(): Array<String> \n fun getAttribute(qualifiedName:
String): String? \n
 fun getAttributeNS(namespace: String?, localName: String): String? \n fun setAttribute(qualifiedName: String,
value: String) \n fun setAttributeNS(namespace: String?, qualifiedName: String, value: String) \n fun
removeAttribute(qualifiedName: String) \n fun removeAttributeNS(namespace: String?, localName: String) \n
fun hasAttribute(qualifiedName: String): Boolean \n fun hasAttributeNS(namespace: String?, localName: String):
Boolean \n fun getAttributeNode(qualifiedName: String): Attr? \n fun getAttributeNodeNS(namespace: String?,
localName: String): Attr? \n fun setAttributeNode(attr: Attr): Attr? \n fun setAttributeNodeNS(attr: Attr): Attr? \n
fun removeAttributeNode(attr: Attr): Attr \n fun attachShadow(init: ShadowRootInit): ShadowRoot \n fun
closest(selectors: String): Element? \n fun matches(selectors: String): Boolean \n fun
webkitMatchesSelector(selectors: String): Boolean \n fun getElementsByTagName(qualifiedName: String):
HTMLCollection \n
 fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection \n fun
getElementsByTagName(className: String): HTMLCollection \n fun insertAdjacentElement(where: String,
element: Element): Element? \n fun insertAdjacentText(where: String, data: String) \n fun getClientRects():
Array<DOMRect> \n fun getBoundingClientRect(): DOMRect \n fun scrollIntoView() \n fun
scrollIntoView(arg: dynamic) \n fun scroll(options: ScrollToOptions = definedExternally) \n fun scroll(x: Double,
y: Double) \n fun scrollTo(options: ScrollToOptions = definedExternally) \n fun scrollTo(x: Double, y: Double) \n
 fun scrollBy(options: ScrollToOptions = definedExternally) \n fun scrollBy(x: Double, y: Double) \n fun
insertAdjacentHTML(position: String, text: String) \n fun setPointerCapture(pointerId: Int) \n fun
releasePointerCapture(pointerId: Int) \n fun hasPointerCapture(pointerId: Int): Boolean \n fun requestFullscreen():
Promise<Unit> \n \n companion
object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE:
Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n public external interface
ShadowRootInit { \n var mode: ShadowRootMode? \n} \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun ShadowRootInit(mode:
ShadowRootMode?): ShadowRootInit
{ \n val o = js("({})") \n o["mode"] = mode \n return o \n} \n \n /** \n * Exposes the JavaScript
[NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to Kotlin \n * \n public external

```



```

abstract class NamedNodeMap : ItemArrayLike<Attr> {
 fun getNamedItemNS(namespace: String?, localName: String): Attr?
 fun setNamedItem(attr: Attr): Attr?
 fun setNamedItemNS(attr: Attr): Attr?
 fun removeNamedItem(qualifiedName: String): Attr
 fun removeNamedItemNS(namespace: String?, localName: String): Attr
 override fun item(index: Int): Attr?
 fun getNamedItem(qualifiedName: String): Attr?
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun NamedNodeMap.get(index: Int): Attr? = asDynamic()[index]
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun NamedNodeMap.get(qualifiedName: String): Attr? = asDynamic()[qualifiedName]
/** Exposes the JavaScript [Attr](https://developer.mozilla.org/en/docs/Web/API/Attr) to Kotlin */
public external abstract class Attr : Node {
 open val namespaceURI: String?
 open val prefix: String?
 open val localName: String
 open val name: String
 open var value: String
 open val ownerElement: Element?
 open val specified: Boolean
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}
/** Exposes the JavaScript [CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin */
public external abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {
 open var data: String
 open val length: Int
 fun substringData(offset: Int, count: Int): String
 fun appendData(data: String)
 fun insertData(offset: Int, data: String)
 fun deleteData(offset: Int, count: Int)
 fun replaceData(offset: Int, count: Int, data: String)
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}
/** Exposes the JavaScript [Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin */
public external open class Text(data: String = definedExternally) : CharacterData, Slotable, GeometryUtils {
 open val wholeText: String
 override val assignedSlot: HTMLSlotElement?
 override val previousElementSibling: Element?
 override val nextElementSibling: Element?
 fun splitText(offset: Int): Text
 override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
 override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
 override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
 override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint
 override fun before(vararg nodes: dynamic)
 override fun after(vararg nodes: dynamic)
 override fun replaceWith(vararg nodes: dynamic)
 override fun remove()
 companion

```

```

object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin

public external open class CDATASection : Text {
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin

public external abstract class ProcessingInstruction : CharacterData, LinkStyle, UnionElementOrProcessingInstruction {
 open val target: String
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin

public external open class Comment(data: String = definedExternally) : CharacterData {
 override val previousElementSibling: Element?
 override val nextElementSibling: Element?
 override fun before(vararg nodes: dynamic)
 override fun after(vararg nodes: dynamic)
 override fun replaceWith(vararg nodes: dynamic)
 override fun remove()

 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin

public external open class Range {
 open val startContainer: Node
 open val startOffset: Int
 open val endContainer: Node
 open val

```

```

endOffset: Int\n open val collapsed: Boolean\n
open val commonAncestorContainer: Node\n fun setStart(node: Node, offset: Int)\n fun setEnd(node: Node,
offset: Int)\n fun setStartBefore(node: Node)\n fun setStartAfter(node: Node)\n fun setEndBefore(node:
Node)\n fun setEndAfter(node: Node)\n fun collapse(toStart: Boolean = definedExternally)\n fun
selectNode(node: Node)\n fun selectNodeContents(node: Node)\n fun compareBoundaryPoints(how: Short,
sourceRange: Range): Short\n fun deleteContents()\n fun extractContents(): DocumentFragment\n fun
cloneContents(): DocumentFragment\n fun insertNode(node: Node)\n fun surroundContents(newParent: Node)\n
fun cloneRange(): Range\n fun detach()\n fun isPointInRange(node: Node, offset: Int): Boolean\n fun
comparePoint(node: Node, offset: Int): Short\n fun intersectsNode(node: Node): Boolean\n fun getClientRects():
Array<DOMRect>\n fun getBoundingClientRect(): DOMRect\n fun createContextualFragment(fragment:
String): DocumentFragment\n\n
companion object {\n val START_TO_START: Short\n val START_TO_END: Short\n val
END_TO_END: Short\n val END_TO_START: Short\n }\n\n/**\n * Exposes the JavaScript
[NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin\n */\n\npublic external abstract
class NodeIterator {\n open val root: Node\n open val referenceNode: Node\n open val
pointerBeforeReferenceNode: Boolean\n open val whatToShow: Int\n open val filter: NodeFilter?\n fun
nextNode(): Node?\n fun previousNode(): Node?\n fun detach()\n }\n\n/**\n * Exposes the JavaScript
[TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin\n */\n\npublic external abstract
class TreeWalker {\n open val root: Node\n open val whatToShow: Int\n open val filter: NodeFilter?\n open
var currentNode: Node\n fun parentNode(): Node?\n fun firstChild(): Node?\n fun lastChild(): Node?\n fun
previousSibling():
Node?\n fun nextSibling(): Node?\n fun previousNode(): Node?\n fun nextNode(): Node?\n }\n\n/**\n *
Exposes the JavaScript [NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin\n
*/\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external interface NodeFilter {\n
fun acceptNode(node: Node): Short\n\n companion object {\n val FILTER_ACCEPT: Short\n val
FILTER_REJECT: Short\n val FILTER_SKIP: Short\n val SHOW_ALL: Int\n val
SHOW_ELEMENT: Int\n val SHOW_ATTRIBUTE: Int\n val SHOW_TEXT: Int\n val
SHOW_CDATA_SECTION: Int\n val SHOW_ENTITY_REFERENCE: Int\n val SHOW_ENTITY: Int\n
val SHOW_PROCESSING_INSTRUCTION: Int\n val SHOW_COMMENT: Int\n val
SHOW_DOCUMENT: Int\n val SHOW_DOCUMENT_TYPE: Int\n val
SHOW_DOCUMENT_FRAGMENT: Int\n val SHOW_NOTATION: Int\n }\n }\n\n/**\n * Exposes the
JavaScript [DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList)
to Kotlin\n */\n\npublic external abstract class DOMTokenList : ItemArrayLike<String> {\n open var value:
String\n fun contains(token: String): Boolean\n fun add(vararg tokens: String)\n fun remove(vararg tokens:
String)\n fun toggle(token: String, force: Boolean = definedExternally): Boolean\n fun replace(token: String,
newToken: String)\n fun supports(token: String): Boolean\n override fun item(index: Int):
String?\n }\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun DOMTokenList.get(index:
Int): String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin\n */\n\npublic
external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {\n open val x: Double\n
open val y: Double\n open val z: Double\n open val w: Double\n fun matrixTransform(matrix:
DOMMatrixReadOnly): DOMPoint\n }\n\n/**\n * Exposes the JavaScript
[DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin\n */\n\npublic external open class
DOMPoint : DOMPointReadOnly {\n constructor(point: DOMPointInit)\n constructor(x: Double =
definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)\n
override var x: Double\n override var y: Double\n override var z: Double\n override var w:
Double\n }\n\n/**\n * Exposes the JavaScript

```

```

[DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin\n */\npublic external
interface DOMPointInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) =
definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n set(value) =
definedExternally\n var z: Double? /* = 0.0 */\n get() = definedExternally\n set(value) =
definedExternally\n
 var w: Double? /* = 1.0 */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMPointInit(x: Double? = 0.0, y:
Double? = 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {\n val o = js(\"({})\")\n o[\"x\"] = x\n
o[\"y\"] = y\n o[\"z\"] = z\n o[\"w\"] = w\n return o\n}\n\n/**\n * Exposes the JavaScript
[DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect) to Kotlin\n */\npublic external open class
DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally,
height: Double = definedExternally) : DOMRectReadOnly {\n override var x: Double\n override var y: Double\n
override var width: Double\n override var height: Double\n}\n\n/**\n * Exposes the JavaScript
[DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin\n */\npublic
external
open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {\n open val x: Double\n
open val y: Double\n open val width: Double\n open val height: Double\n open val top: Double\n open val
right: Double\n open val bottom: Double\n open val left: Double\n}\n\npublic external interface DOMRectInit
{\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var y:
Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var width: Double? /*
= 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /* = 0.0 */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMRectInit(x: Double? = 0.0, y:
Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit
{\n val o = js(\"({})\")\n o[\"x\"] = x\n o[\"y\"] = y\n o[\"width\"] = width\n o[\"height\"] = height\n
return o\n}\n\npublic external interface DOMRectList : ItemArrayLike<DOMRect> {\n override fun item(index:
Int): DOMRect?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMRectList.get(index: Int):
DOMRect? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad) to Kotlin\n */\npublic external open class
DOMQuad {\n constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3:
DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)\n constructor(rect: DOMRectInit)\n
open val p1: DOMPoint\n open val p2: DOMPoint\n open val p3: DOMPoint\n open val p4: DOMPoint\n
open val bounds: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript
[DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly)
to Kotlin\n */\npublic external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {\n open val
a: Double\n open val b: Double\n open val c: Double\n open val d: Double\n open val e: Double\n open val
f: Double\n open val m11: Double\n open val m12: Double\n open val m13: Double\n open val m14:
Double\n open val m21: Double\n open val m22: Double\n open val m23: Double\n open val m24: Double\n
open val m31: Double\n open val m32: Double\n open val m33: Double\n open val m34: Double\n open val
m41: Double\n open val m42: Double\n open val m43: Double\n open val m44: Double\n open val is2D:
Boolean\n open val isIdentity: Boolean\n fun translate(tx: Double, ty: Double, tz: Double = definedExternally):
DOMMatrix\n fun scale(scale: Double, originX: Double = definedExternally, originY: Double =
definedExternally): DOMMatrix\n fun scale3d(scale: Double,
originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally):
DOMMatrix\n fun scaleNonUniform(scaleX: Double, scaleY: Double = definedExternally, scaleZ: Double =
definedExternally, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double =

```

definedExternally): DOMMatrix\n fun rotate(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n fun rotateFromVector(x: Double, y: Double): DOMMatrix\n fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix\n fun skewX(sx: Double): DOMMatrix\n fun skewY(sy: Double): DOMMatrix\n fun multiply(other: DOMMatrix): DOMMatrix\n fun flipX(): DOMMatrix\n fun flipY(): DOMMatrix\n fun inverse(): DOMMatrix\n fun transformPoint(point: DOMPointInit = definedExternally): DOMPoint\n fun toFloat32Array(): Float32Array\n fun toFloat64Array(): Float64Array\n}\n\n/\*\*\n \*

Exposes the JavaScript [DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin\n \*\npublic external open class DOMMatrix() : DOMMatrixReadOnly {\n constructor(transformList: String)\n constructor(other: DOMMatrixReadOnly)\n constructor(array32: Float32Array)\n constructor(array64: Float64Array)\n constructor(numberSequence: Array<Double>)\n override var a: Double\n override var b: Double\n override var c: Double\n override var d: Double\n override var e: Double\n override var f: Double\n override var m11: Double\n override var m12: Double\n override var m13: Double\n override var m14: Double\n override var m21: Double\n override var m22: Double\n override var m23: Double\n override var m24: Double\n override var m31: Double\n override var m32: Double\n override var m33: Double\n override var m34: Double\n override var m41: Double\n override var m42: Double\n override var m43: Double\n

override var m44: Double\n fun multiplySelf(other: DOMMatrix): DOMMatrix\n fun preMultiplySelf(other: DOMMatrix): DOMMatrix\n fun translateSelf(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix\n fun scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n fun scale3dSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n fun scaleNonUniformSelf(scaleX: Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n fun rotateSelf(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n fun rotateFromVectorSelf(x: Double, y: Double): DOMMatrix\n fun rotateAxisAngleSelf(x: Double, y: Double, z: Double, angle: Double):

DOMMatrix\n fun skewXSelf(sx: Double): DOMMatrix\n fun skewYSelf(sy: Double): DOMMatrix\n fun invertSelf(): DOMMatrix\n fun setMatrixValue(transformList: String): DOMMatrix\n}\n\npublic external interface ScrollOptions {\n var behavior: ScrollBehavior? /\* = ScrollBehavior.AUTO \*/\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE\_REFERENCE\", \"INVISIBLE\_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollOptions(behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {\n val o = js(\"({})\")\n o[\"behavior\"] = behavior\n return o\n}\n\n/\*\*\n \*

[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin\n \*\npublic external interface ScrollToOptions : ScrollOptions {\n var left: Double?\n get() = definedExternally\n set(value) = definedExternally\n var top: Double?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE\_REFERENCE\", \"INVISIBLE\_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollToOptions(left: Double? = undefined, top: Double? = undefined, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollToOptions {\n val o = js(\"({})\")\n o[\"left\"] = left\n o[\"top\"] = top\n o[\"behavior\"] = behavior\n return o\n}\n\n/\*\*\n \*

Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin\n \*\npublic external abstract class MediaQueryList : EventTarget {\n open val media: String\n open val matches: Boolean\n open var onchange: ((Event) -> dynamic)?\n fun addListener(listener: EventListener?)\n fun addListener(listener: ((Event) -> Unit)?)\n fun removeListener(listener: EventListener?)\n fun removeListener(listener: ((Event) -> Unit)?)\n}\n\n/\*\*\n \*

[MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent)

```

to Kotlin\n */\npublic external open class MediaQueryListEvent(type: String, eventInitDict:
MediaQueryListEventInit = definedExternally) : Event {\n open val media: String\n open val matches:
Boolean\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
MediaQueryListEventInit : EventInit {\n var media: String? /* = \"\" */\n get() = definedExternally\n
set(value) = definedExternally\n var matches: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaQueryListEventInit(media:
String? = \"\", matches: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): MediaQueryListEventInit {\n val o = js(\"({})\")\n
 o[\"media\"] = media\n o[\"matches\"] = matches\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] =
cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin\n */\npublic external abstract class
Screen {\n open val availWidth: Int\n open val availHeight: Int\n open val width: Int\n open val height: Int\n
 open val colorDepth: Int\n open val pixelDepth: Int\n}\n\n/**\n * Exposes the JavaScript
[CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin\n */\npublic external
abstract class CaretPosition {\n open val offsetNode: Node\n open val offset: Int\n fun getClientRect():
DOMRect?}\n}\n\npublic external interface ScrollIntoViewOptions : ScrollOptions {\n var block:
ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n get() = definedExternally\n set(value) =
definedExternally\n var inline:
ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollIntoViewOptions(block:
ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, inline: ScrollLogicalPosition? =
ScrollLogicalPosition.CENTER, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollIntoViewOptions {\n
 val o = js(\"({})\")\n o[\"block\"] = block\n o[\"inline\"] = inline\n o[\"behavior\"] = behavior\n
 return o\n}\n\npublic external interface BoxQuadOptions {\n var box: CSSBoxType? /* = CSSBoxType.BORDER */\n
 get() = definedExternally\n set(value) = definedExternally\n var relativeTo: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun BoxQuadOptions(box:
CSSBoxType? = CSSBoxType.BORDER, relativeTo: dynamic = undefined): BoxQuadOptions {\n val o =
js(\"({})\")\n o[\"box\"] = box\n o[\"relativeTo\"] = relativeTo\n return o\n}\n\npublic external interface
ConvertCoordinateOptions {\n var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */\n get() =
definedExternally\n set(value) = definedExternally\n var toBox: CSSBoxType? /* = CSSBoxType.BORDER
*/\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConvertCoordinateOptions(fromBox:
CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER):
ConvertCoordinateOptions {\n val o = js(\"({})\")\n o[\"fromBox\"] = fromBox\n o[\"toBox\"] = toBox\n
 return o\n}\n\n/**\n * Exposes the JavaScript
[GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin\n */\npublic external
interface GeometryUtils {\n fun getBoxQuads(options: BoxQuadOptions = definedExternally):
Array<DOMQuad>\n fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions = definedExternally): DOMQuad\n fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions =
definedExternally): DOMPoint\n}\n\n/**\n * Exposes the JavaScript
[Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin\n */\npublic external abstract class Touch

```

```

{\n open val identifier: Int\n open val target: EventTarget\n open val screenX: Int\n open val screenY: Int\n
open val clientX: Int\n open val clientY: Int\n open val pageX: Int\n open val pageY: Int\n open val region:
String?\n}\n\npublic external abstract class TouchList : ItemArrayLike<Touch> {\n override fun item(index:
Int): Touch?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun TouchList.get(index: Int):
Touch? = asDynamic()[index]\n\npublic external open class TouchEvent : UIEvent {\n open val touches:
TouchList\n open val targetTouches: TouchList\n open val changedTouches: TouchList\n open val altKey:
Boolean\n open val metaKey: Boolean\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n\n
companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET:
Short\n val BUBBLING_PHASE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin\n *\n\npublic external open class
Image(width: Int = definedExternally, height: Int = definedExternally) : HTMLImageElement {\n override var
onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel:
((Event)
-> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough: ((Event) ->
dynamic)?\n override var onchange: ((Event) -> dynamic)?\n override var onclick: ((MouseEvent) ->
dynamic)?\n override var onclose: ((Event) -> dynamic)?\n override var oncontextmenu: ((MouseEvent) ->
dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var ondblclick: ((MouseEvent) ->
dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) ->
dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var ondragexit: ((DragEvent) ->
dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover: ((DragEvent) ->
dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) ->
dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) ->
dynamic)?\n
override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int, Any?) ->
dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) ->
dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup:
((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata:
((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend:
((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown:
((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var
onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n
override var onmouseout: ((MouseEvent)
-> dynamic)?\n override var onmouseover: ((MouseEvent) -> dynamic)?\n override var onmouseup:
((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) -> dynamic)?\n override var onpause:
((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n override var onplaying: ((Event) ->
dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n override var onratechange: ((Event) ->
dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var onresize: ((Event) -> dynamic)?\n
override var onscroll: ((Event) -> dynamic)?\n override var onseeked: ((Event) -> dynamic)?\n override var
onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) -> dynamic)?\n override var onshow:
((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n override var onsubmit: ((Event) ->
dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var ontimeupdate: ((Event) ->
dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var onvolumechange: ((Event) ->
dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var ongotpointercapture: ((PointerEvent)
-> dynamic)?\n override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n override var onpointerdown:
((PointerEvent) -> dynamic)?\n override var onpointermove: ((PointerEvent) -> dynamic)?\n override var
onpointerup: ((PointerEvent) -> dynamic)?\n override var onpointercancel: ((PointerEvent) -> dynamic)?\n
}

```

```

override var onpointerover: ((PointerEvent) -> dynamic)?\n override var onpointerout: ((PointerEvent) ->
dynamic)?\n override var onpointerenter: ((PointerEvent) -> dynamic)?\n override var onpointerleave:
((PointerEvent) -> dynamic)?\n override var oncopy: ((ClipboardEvent) -> dynamic)?\n override var oncut:
((ClipboardEvent) -> dynamic)?\n override var onpaste: ((ClipboardEvent) -> dynamic)?\n override var
contentEditable:
String\n override val isContentEditable: Boolean\n override val style: CSSStyleDeclaration\n override val
children: HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild:
Element?\n override val childElementCount: Int\n override val previousElementSibling: Element?\n override
val nextElementSibling: Element?\n override val assignedSlot: HTMLSlotElement?\n override fun
prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun
querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n
override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override fun
replaceWith(vararg nodes: dynamic)\n override fun remove()\n override fun getBoxQuads(options:
BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions
/* = definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\npublic external open class
Audio(src: String = definedExternally) : HTMLAudioElement {\n override var onabort: ((Event) -> dynamic)?\n
override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel: ((Event) -> dynamic)?\n override var
oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough: ((Event) -> dynamic)?\n override var
onchange: ((Event) -> dynamic)?\n override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose:
((Event) -> dynamic)?\n override var oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange:
((Event) -> dynamic)?\n override var ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag:
((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter:
((DragEvent) -> dynamic)?\n override var ondragexit:
((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover:
((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop:
((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied:
((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String,
Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput:
((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown:
((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var
onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var
onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var
onloadend:
((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown:
((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var

```



```

onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n
override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) ->
dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) ->
dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var
onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked:
((Event) ->
dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) -> dynamic)?\n
override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n override var
onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var ontimeupdate:
((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var onvolumechange: ((Event)
-> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var ongotpointercapture:
((PointerEvent) -> dynamic)?\n override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n override var
onpointerdown: ((PointerEvent) -> dynamic)?\n override var onpointermove: ((PointerEvent) -> dynamic)?\n
override var onpointerup: ((PointerEvent) -> dynamic)?\n override var onpointercancel: ((PointerEvent) ->
dynamic)?\n override var onpointerover: ((PointerEvent) -> dynamic)?\n override var onpointerout:
((PointerEvent) -> dynamic)?\n override var onpointerenter: ((PointerEvent) -> dynamic)?\n override var
onpointerleave: ((PointerEvent) -> dynamic)?\n override var oncopy: ((ClipboardEvent) -> dynamic)?\n override
var oncut: ((ClipboardEvent) -> dynamic)?\n override var onpaste: ((ClipboardEvent) -> dynamic)?\n override
var contentEditable: String\n override val isContentEditable: Boolean\n override val style:
CSSStyleDeclaration\n override val children: HTMLCollection\n override val firstElementChild: Element?\n
override val lastElementChild: Element?\n override val childElementCount: Int\n override val
previousElementSibling: Element?\n override val nextElementSibling: Element?\n override val assignedSlot:
HTMLSlotElement?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes:
dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors:
String): NodeList\n
 override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override fun
replaceWith(vararg nodes: dynamic)\n override fun remove()\n override fun getBoxQuads(options:
BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object {\n val
NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA:
Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA:
Short\n val HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin\n */\npublic external open class

```

```

Option(text: String = definedExternally, value: String
= definedExternally, defaultSelected: Boolean = definedExternally, selected: Boolean = definedExternally) :
HTMLInputElement {
 override var onabort: ((Event) -> dynamic)?
 override var onblur: ((FocusEvent) ->
dynamic)?
 override var onCancel: ((Event) -> dynamic)?
 override var oncanplay: ((Event) -> dynamic)?
 override var oncanplaythrough: ((Event) -> dynamic)?
 override var onChange: ((Event) -> dynamic)?
 override var onclick: ((MouseEvent) -> dynamic)?
 override var onClose: ((Event) -> dynamic)?
 override var oncontextmenu: ((MouseEvent) -> dynamic)?
 override var oncuechange: ((Event) -> dynamic)?
 override var ondblclick: ((MouseEvent) -> dynamic)?
 override var ondrag: ((DragEvent) -> dynamic)?
 override var
ondragend: ((DragEvent) -> dynamic)?
 override var ondragenter: ((DragEvent) -> dynamic)?
 override var
ondragexit: ((DragEvent) -> dynamic)?
 override var ondragleave: ((DragEvent) -> dynamic)?
 override var ondragover: ((DragEvent) -> dynamic)?
 override var ondragstart: ((DragEvent) -> dynamic)?
 override var ondrop: ((DragEvent) -> dynamic)?
 override var ondurationchange: ((Event) -> dynamic)?
 override var onemptied: ((Event) -> dynamic)?
 override var onended: ((Event) -> dynamic)?
 override var
onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
 override var onfocus: ((FocusEvent) -> dynamic)?
 override var oninput: ((InputEvent) -> dynamic)?
 override var oninvalid: ((Event) -> dynamic)?
 override var
onkeydown: ((KeyboardEvent) -> dynamic)?
 override var onkeypress: ((KeyboardEvent) -> dynamic)?
 override var onkeyup: ((KeyboardEvent) -> dynamic)?
 override var onload: ((Event) -> dynamic)?
 override
var onloadeddata: ((Event) -> dynamic)?
 override var onloadedmetadata: ((Event) -> dynamic)?
 override var
onloadend: ((Event) -> dynamic)?
 override var onloadstart: ((ProgressEvent) -> dynamic)?
 override var
onmousedown: ((MouseEvent) -> dynamic)?
 override var onmouseenter: ((MouseEvent) ->
dynamic)?
 override var onmouseleave: ((MouseEvent) -> dynamic)?
 override var onmousemove:
((MouseEvent) -> dynamic)?
 override var onmouseout: ((MouseEvent) -> dynamic)?
 override var
onmouseover: ((MouseEvent) -> dynamic)?
 override var onmouseup: ((MouseEvent) -> dynamic)?
 override
var onwheel: ((WheelEvent) -> dynamic)?
 override var onpause: ((Event) -> dynamic)?
 override var onplay:
((Event) -> dynamic)?
 override var onplaying: ((Event) -> dynamic)?
 override var onprogress:
((ProgressEvent) -> dynamic)?
 override var onratechange: ((Event) -> dynamic)?
 override var onreset:
((Event) -> dynamic)?
 override var onresize: ((Event) -> dynamic)?
 override var onscroll: ((Event) ->
dynamic)?
 override var onseeked: ((Event) -> dynamic)?
 override var onseeking: ((Event) -> dynamic)?
 override var onselect: ((Event)
-> dynamic)?
 override var onshow: ((Event) -> dynamic)?
 override var onstalled: ((Event) -> dynamic)?
 override var
onsubmit: ((Event) -> dynamic)?
 override var onsuspend: ((Event) -> dynamic)?
 override var
ontimeupdate: ((Event) -> dynamic)?
 override var ontoggle: ((Event) -> dynamic)?
 override var
onvolumechange: ((Event) -> dynamic)?
 override var onwaiting: ((Event) -> dynamic)?
 override var
ongotpointercapture: ((PointerEvent) -> dynamic)?
 override var onlostpointercapture: ((PointerEvent) ->
dynamic)?
 override var onpointerdown: ((PointerEvent) -> dynamic)?
 override var onpointermove:
((PointerEvent) -> dynamic)?
 override var onpointerup: ((PointerEvent) -> dynamic)?
 override var
onpointercancel: ((PointerEvent) -> dynamic)?
 override var onpointerover: ((PointerEvent) -> dynamic)?
 override var
onpointerout: ((PointerEvent) -> dynamic)?
 override var onpointerenter: ((PointerEvent) ->
dynamic)?
 override var onpointerleave: ((PointerEvent) -> dynamic)?
 override var oncopy: ((ClipboardEvent) ->
dynamic)?
 override var oncut: ((ClipboardEvent) -> dynamic)?
 override var onpaste: ((ClipboardEvent) ->
dynamic)?
 override var contentEditable: String
 override val isContentEditable: Boolean
 override val
style: CSSStyleDeclaration
 override val children: HTMLCollection
 override val firstElementChild:
Element?
 override val lastElementChild: Element?
 override val childElementCount: Int
 override val
previousElementSibling: Element?
 override val nextElementSibling: Element?
 override val assignedSlot:
HTMLSlotElement?
 override fun prepend(vararg nodes: dynamic)
 override fun append(vararg nodes:
dynamic)
 override fun querySelector(selectors: String): Element?
 override fun querySelectorAll(selectors:
String): NodeList
 override fun before(vararg nodes: dynamic)
 override fun after(vararg nodes: dynamic)

```

```

 override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n override fun
getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun
convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally
*/): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options:
ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertPointFromNode(point:
DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\npublic external interface
UnionElementOrHTMLCollection\n\npublic external interface UnionElementOrRadioNodeList\n\npublic external
interface UnionHTMLOptGroupElementOrHTMLOptionElement\n\npublic external interface
UnionAudioTrackOrTextTrackOrVideoTrack\n\npublic external interface UnionElementOrMouseEvent\n\npublic
external interface UnionMessagePortOrWindowProxy\n\npublic external interface MediaProvider\n\npublic
external interface RenderingContext\n\npublic external interface HTMLOrSVGImageElement :
CanvasImageSource\n\npublic external interface CanvasImageSource : ImageBitmapSource\n\npublic
external interface ImageBitmapSource\n\npublic external interface HTMLOrSVGScriptElement\n\n/* please, don't
implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface DocumentReadyState {\n companion object\n}\n\npublic inline val
DocumentReadyState.Companion.LOADING: DocumentReadyState get() =
"loading".asDynamic().unsafeCast<DocumentReadyState>()\n\npublic inline val
DocumentReadyState.Companion.INTERACTIVE: DocumentReadyState get() =
"interactive".asDynamic().unsafeCast<DocumentReadyState>()\n\npublic inline val
DocumentReadyState.Companion.COMPLETE: DocumentReadyState get() =
"complete".asDynamic().unsafeCast<DocumentReadyState>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanPlayTypeResult {\n companion object\n}\n\npublic inline val
CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() =
"".asDynamic().unsafeCast<CanPlayTypeResult>()\n\npublic
inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() =
"maybe".asDynamic().unsafeCast<CanPlayTypeResult>()\n\npublic inline val
CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() =
"probably".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface TextTrackMode {\n companion object\n}\n\npublic inline val TextTrackMode.Companion.DISABLED:
TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.HIDDEN: TextTrackMode get() =
"hidden".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.SHOWING: TextTrackMode get() =
"showing".asDynamic().unsafeCast<TextTrackMode>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic

```

```

external interface TextTrackKind {
 companion object
}
public inline val
TextTrackKind.Companion.SUBTITLES: TextTrackKind get() =
 "subtitles".asDynamic().unsafeCast<TextTrackKind>()
public inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
 "captions".asDynamic().unsafeCast<TextTrackKind>()
public inline val
TextTrackKind.Companion.DESCRPTIONS: TextTrackKind get() =
 "descriptions".asDynamic().unsafeCast<TextTrackKind>()
public inline val
TextTrackKind.Companion.CHAPTERS: TextTrackKind get() =
 "chapters".asDynamic().unsafeCast<TextTrackKind>()
public inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
 "metadata".asDynamic().unsafeCast<TextTrackKind>()
/* please, don't implement this interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external
interface SelectionMode {
 companion object
}
public inline val SelectionMode.Companion.SELECT:
SelectionMode get()
 = "select".asDynamic().unsafeCast<SelectionMode>()
public inline val SelectionMode.Companion.START:
SelectionMode get() = "start".asDynamic().unsafeCast<SelectionMode>()
public inline val
SelectionMode.Companion.END: SelectionMode get() =
 "end".asDynamic().unsafeCast<SelectionMode>()
public inline val SelectionMode.Companion.PRESERVE:
SelectionMode get() = "preserve".asDynamic().unsafeCast<SelectionMode>()
/* please, don't implement this
interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public
external interface CanvasFillRule {
 companion object
}
public inline val
CanvasFillRule.Companion.NONZERO: CanvasFillRule get() =
 "nonzero".asDynamic().unsafeCast<CanvasFillRule>()
public inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() =
 "evenodd".asDynamic().unsafeCast<CanvasFillRule>()
/* please, don't implement this interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public
external interface ImageSmoothingQuality {
 companion object
}
public inline val
ImageSmoothingQuality.Companion.LOW: ImageSmoothingQuality get() =
 "low".asDynamic().unsafeCast<ImageSmoothingQuality>()
public inline val
ImageSmoothingQuality.Companion.MEDIUM: ImageSmoothingQuality get() =
 "medium".asDynamic().unsafeCast<ImageSmoothingQuality>()
public inline val
ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() =
 "high".asDynamic().unsafeCast<ImageSmoothingQuality>()
/* please, don't implement this interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external
interface CanvasLineCap {
 companion object
}
public inline val CanvasLineCap.Companion.BUTT:
CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()
public inline val
CanvasLineCap.Companion.ROUND: CanvasLineCap get() =
 "round".asDynamic().unsafeCast<CanvasLineCap>()
public inline val CanvasLineCap.Companion.SQUARE:
CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()
/* please, don't implement this
interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public
external interface CanvasLineJoin {
 companion object
}
public inline val
CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() =
 "round".asDynamic().unsafeCast<CanvasLineJoin>()
public inline val CanvasLineJoin.Companion.BEVEL:
CanvasLineJoin get() = "bevel".asDynamic().unsafeCast<CanvasLineJoin>()
public inline val
CanvasLineJoin.Companion.MITER: CanvasLineJoin get() =
 "miter".asDynamic().unsafeCast<CanvasLineJoin>()
/* please, don't implement this interface!
*/
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external
interface CanvasTextAlign {
 companion object
}

```

```

CanvasTextAlign get() = \"start\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.END:
CanvasTextAlign get() = \"end\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.LEFT: CanvasTextAlign get() =
\"left\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val CanvasTextAlign.Companion.RIGHT:
CanvasTextAlign get() = \"right\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() =
\"center\".asDynamic().unsafeCast<CanvasTextAlign>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CanvasTextBaseline {\n\n\ncompanion object\n\n}\n\n\npublic inline val CanvasTextBaseline.Companion.TOP:
CanvasTextBaseline get() = \"top\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() =
\"hanging\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.MIDDLE:
CanvasTextBaseline get() = \"middle\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() =
\"alphabetic\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() =
\"ideographic\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() =
\"bottom\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CanvasDirection {\n\n\ncompanion object\n\n}\n\n\npublic inline val CanvasDirection.Companion.LTR:
CanvasDirection get() = \"ltr\".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val
CanvasDirection.Companion.RTL: CanvasDirection get() =
\"rtl\".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val CanvasDirection.Companion.INHERIT:
CanvasDirection get() = \"inherit\".asDynamic().unsafeCast<CanvasDirection>()\n\n\n/* please, don't implement this
interface! *\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ScrollRestoration {\n\n\ncompanion object\n\n}\n\n\npublic inline val
ScrollRestoration.Companion.AUTO: ScrollRestoration get() =
\"auto\".asDynamic().unsafeCast<ScrollRestoration>()\n\npublic inline val
ScrollRestoration.Companion.MANUAL: ScrollRestoration get() =
\"manual\".asDynamic().unsafeCast<ScrollRestoration>()\n\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ImageOrientation {\n\n\ncompanion object\n\n}\n\n\npublic inline val ImageOrientation.Companion.NONE:
ImageOrientation get() = \"none\".asDynamic().unsafeCast<ImageOrientation>()\n\npublic inline val
ImageOrientation.Companion.FLIPY: ImageOrientation get() =
\"flipY\".asDynamic().unsafeCast<ImageOrientation>()\n\n\n\n/*
please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface PremultiplyAlpha {\n\n\ncompanion object\n\n}\n\n\npublic inline val PremultiplyAlpha.Companion.NONE:
PremultiplyAlpha get() = \"none\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
\"premultiply\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() =
\"default\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\n\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ColorSpaceConversion {\n\n\ncompanion object\n\n}\n\n\npublic inline val

```

```

ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() =
\"none\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\npublic inline val
ColorSpaceConversion.Companion.DEFAULT:
ColorSpaceConversion get() = \"default\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\n/* please, don't
implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ResizeQuality {\n companion object\n}\n\npublic inline val ResizeQuality.Companion.PIXELATED:
ResizeQuality get() = \"pixelated\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.LOW: ResizeQuality get() =
\"low\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.MEDIUM:
ResizeQuality get() = \"medium\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.HIGH: ResizeQuality get() = \"high\".asDynamic().unsafeCast<ResizeQuality>()\n\n/*
please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface BinaryType {\n companion object\n}\n\npublic inline
val BinaryType.Companion.BLOB: BinaryType get() = \"blob\".asDynamic().unsafeCast<BinaryType>()\n\npublic
inline val BinaryType.Companion.ARRAYBUFFER: BinaryType get() =
\"arraybuffer\".asDynamic().unsafeCast<BinaryType>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface WorkerType {\n companion object\n}\n\npublic inline val WorkerType.Companion.CLASSIC:
WorkerType get() = \"classic\".asDynamic().unsafeCast<WorkerType>()\n\npublic inline val
WorkerType.Companion.MODULE: WorkerType get() =
\"module\".asDynamic().unsafeCast<WorkerType>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ShadowRootMode {\n companion object\n}\n\npublic inline val ShadowRootMode.Companion.OPEN:
ShadowRootMode get() = \"open\".asDynamic().unsafeCast<ShadowRootMode>()\n\npublic inline val
ShadowRootMode.Companion.CLOSED:
ShadowRootMode get() = \"closed\".asDynamic().unsafeCast<ShadowRootMode>()\n\n/* please, don't implement
this interface! *\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ScrollBehavior {\n companion object\n}\n\npublic inline val ScrollBehavior.Companion.AUTO:
ScrollBehavior get() = \"auto\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val
ScrollBehavior.Companion.INSTANT: ScrollBehavior get() =
\"instant\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val ScrollBehavior.Companion.SMOOTH:
ScrollBehavior get() = \"smooth\".asDynamic().unsafeCast<ScrollBehavior>()\n\n/* please, don't implement this
interface! *\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ScrollLogicalPosition {\n companion object\n}\n\npublic inline val
ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() =
\"start\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic
inline val ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() =
\"center\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() =
\"end\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() =
\"nearest\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CSSBoxType {\n companion object\n}\n\npublic inline val CSSBoxType.Companion.MARGIN:
CSSBoxType get() = \"margin\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.BORDER: CSSBoxType get() =

```

```

\"border\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.PADDING:
CSSBoxType get() = \"padding\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline
val CSSBoxType.Companion.CONTENT: CSSBoxType get() =
\"content\".asDynamic().unsafeCast<CSSBoxType>()\", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *^\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.fetch\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.files.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript
[Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n\n *^\n\npublic external open class
Headers(init: dynamic = definedExternally) {\n fun append(name: String, value: String)\n fun delete(name:
String)\n fun get(name: String): String?\n fun has(name: String): Boolean\n fun set(name: String, value:
String)\n}\n\n/**\n * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body)
to Kotlin\n\n *^\n\npublic external interface Body {\n val bodyUsed: Boolean\n fun arrayBuffer():
Promise<ArrayBuffer>\n fun blob(): Promise<Blob>\n fun formData(): Promise<FormData>\n fun json():
Promise<Any?>\n fun text(): Promise<String>\n}\n\n/**\n * Exposes the JavaScript
[Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n\n *^\n\npublic external open class
Request(input: dynamic, init: RequestInit = definedExternally) : Body {\n open val method: String\n open val
url: String\n open val headers: Headers\n open val type: RequestType\n open val destination:
RequestDestination\n open val referrer: String\n open val referrerPolicy: dynamic\n open val mode:
RequestMode\n open val credentials: RequestCredentials\n open val cache: RequestCache\n open val redirect:
RequestRedirect\n open val integrity: String\n open val keepalive: Boolean\n override val bodyUsed:
Boolean\n fun clone(): Request\n override
fun arrayBuffer(): Promise<ArrayBuffer>\n override fun blob(): Promise<Blob>\n override fun formData():
Promise<FormData>\n override fun json(): Promise<Any?>\n override fun text(): Promise<String>\n}\n\npublic
external interface RequestInit {\n var method: String?\n get() = definedExternally\n set(value) =
definedExternally\n var headers: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
var body: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var referrer: String?\n
get() = definedExternally\n set(value) = definedExternally\n var referrerPolicy: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var mode: RequestMode?\n get() =
definedExternally\n set(value) = definedExternally\n var credentials: RequestCredentials?\n get() =
definedExternally\n set(value) = definedExternally\n var
cache: RequestCache?\n get() = definedExternally\n set(value) = definedExternally\n var redirect:
RequestRedirect?\n get() = definedExternally\n set(value) = definedExternally\n var integrity: String?\n
get() = definedExternally\n set(value) = definedExternally\n var keepalive: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var window: Any?\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun RequestInit(method: String? =
undefined, headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy:
dynamic = undefined, mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache:
RequestCache? = undefined, redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive:
Boolean? = undefined,
window: Any? = undefined): RequestInit {\n val o = js(\"({})\")\n o[\"method\"] = method\n o[\"headers\"] =
headers\n o[\"body\"] = body\n o[\"referrer\"] = referrer\n o[\"referrerPolicy\"] = referrerPolicy\n
o[\"mode\"] = mode\n o[\"credentials\"] = credentials\n o[\"cache\"] = cache\n o[\"redirect\"] = redirect\n
o[\"integrity\"] = integrity\n o[\"keepalive\"] = keepalive\n o[\"window\"] = window\n return o\n}\n\n/**\n *
Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response) to Kotlin\n\n *^\n\npublic
external open class Response(body: dynamic = definedExternally, init: ResponseInit = definedExternally) : Body
{\n open val type: ResponseType\n open val url: String\n open val redirected: Boolean\n open val status:

```

```

Short\n open val ok: Boolean\n open val statusText: String\n open val headers: Headers\n open val body:
dynamic\n open val trailer: Promise<Headers>\n override val bodyUsed:
Boolean\n fun clone(): Response\n override fun arrayBuffer(): Promise<ArrayBuffer>\n override fun blob():
Promise<Blob>\n override fun formData(): Promise<FormData>\n override fun json(): Promise<Any?>\n
override fun text(): Promise<String>\n\n companion object {\n fun error(): Response\n fun redirect(url:
String, status: Short = definedExternally): Response\n }\n}\n\npublic external interface ResponseInit {\n var
status: Short? /* = 200 */\n get() = definedExternally\n set(value) = definedExternally\n var statusText:
String? /* = \"OK\" */\n get() = definedExternally\n set(value) = definedExternally\n var headers:
dynamic\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ResponseInit(status: Short? = 200,
statusText: String? = \"OK\", headers: dynamic = undefined): ResponseInit
{\n val o = js(\"({})\")\n o[\"status\"] = status\n o[\"statusText\"] = statusText\n o[\"headers\"] = headers\n
return o\n}\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestType {\n companion object\n}\n\npublic inline val RequestType.Companion.EMPTY:
RequestType get() = \"\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.AUDIO: RequestType get() =
\"audio\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.FONT:
RequestType get() = \"font\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.IMAGE: RequestType get() =
\"image\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.SCRIPT:
RequestType get() = \"script\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.STYLE: RequestType get() =
\"style\".asDynamic().unsafeCast<RequestType>()\n\npublic
inline val RequestType.Companion.TRACK: RequestType get() =
\"track\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.VIDEO:
RequestType get() = \"video\".asDynamic().unsafeCast<RequestType>()\n\n/* please, don't implement this
interface! */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface RequestDestination {\n companion object\n}\n\npublic inline val
RequestDestination.Companion.EMPTY: RequestDestination get() =
\"\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.DOCUMENT: RequestDestination get() =
\"document\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.EMBED: RequestDestination get() =
\"embed\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.FONT: RequestDestination get() =
\"font\".asDynamic().unsafeCast<RequestDestination>()\n\npublic
inline val RequestDestination.Companion.IMAGE: RequestDestination get() =
\"image\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MANIFEST: RequestDestination get() =
\"manifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MEDIA: RequestDestination get() =
\"media\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.OBJECT: RequestDestination get() =
\"object\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.REPORT: RequestDestination get() =
\"report\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val

```



```

RequestDestination.Companion.SCRIPT: RequestDestination get() =
 `script`.asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SERVICEMANAGER: RequestDestination get() =
 `serviceworker`.asDynamic().unsafeCast<RequestDestination>()\n\npublic
 inline val RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
 `sharedworker`.asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.STYLE: RequestDestination get() =
 `style`.asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.WORKER: RequestDestination get() =
 `worker`.asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.XSLT: RequestDestination get() =
 `xslt`.asDynamic().unsafeCast<RequestDestination>()\n\n/* please, don't implement this interface!
 *\n\n@JsName(`null`)\n\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface RequestMode {\n companion object\n}\n\npublic inline val RequestMode.Companion.NAVIGATE:
RequestMode get() = `navigate`.asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.SAME_ORIGIN: RequestMode get() = `same-
origin`.asDynamic().unsafeCast<RequestMode>()\n\npublic
 inline val RequestMode.Companion.NO_CORS: RequestMode get() = `no-
cors`.asDynamic().unsafeCast<RequestMode>()\n\npublic inline val RequestMode.Companion.CORS:
RequestMode get() = `cors`.asDynamic().unsafeCast<RequestMode>()\n\n/* please, don't implement this
interface! *\n\n@JsName(`null`)\n\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic
external interface RequestCredentials {\n companion object\n}\n\npublic inline val
RequestCredentials.Companion.OMIT: RequestCredentials get() =
 `omit`.asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = `same-
origin`.asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.INCLUDE: RequestCredentials get() =
 `include`.asDynamic().unsafeCast<RequestCredentials>()\n\n/* please, don't implement this interface!
 *\n\n@JsName(`null`)\n\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic
external interface RequestCache {\n companion object\n}\n\npublic inline val
RequestCache.Companion.DEFAULT: RequestCache get() =
 `default`.asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.NO_STORE:
RequestCache get() = `no-store`.asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.RELOAD: RequestCache get() =
 `reload`.asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.NO_CACHE:
RequestCache get() = `no-cache`.asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.FORCE_CACHE: RequestCache get() = `force-
cache`.asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = `only-if-
cached`.asDynamic().unsafeCast<RequestCache>()\n\n/* please, don't implement this interface!
 *\n\n@JsName(`null`)\n\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface RequestRedirect
 {\n companion object\n}\n\npublic inline val RequestRedirect.Companion.FOLLOW: RequestRedirect get() =
 `follow`.asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val RequestRedirect.Companion.ERROR:
RequestRedirect get() = `error`.asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val
RequestRedirect.Companion.MANUAL: RequestRedirect get() =
 `manual`.asDynamic().unsafeCast<RequestRedirect>()\n\n/* please, don't implement this interface!
 *\n\n@JsName(`null`)\n\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external

```

```

interface ResponseType {
 companion object
}

public inline val ResponseType.Companion.BASIC:
ResponseType get() = "basic".asDynamic().unsafeCast<ResponseType>()

public inline val
ResponseType.Companion.CORS: ResponseType get() =
"cors".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.DEFAULT:
ResponseType get() = "default".asDynamic().unsafeCast<ResponseType>()

public inline
val ResponseType.Companion.ERROR: ResponseType get() =
"error".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.OPAQUE:
ResponseType get() = "opaque".asDynamic().unsafeCast<ResponseType>()

public inline val
ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() =
"opaqueredirect".asDynamic().unsafeCast<ResponseType>()

/*
 * Copyright 2010-2021 JetBrains s.r.o. and
 * Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that
 * can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
// EDIT!

// See github.com/kotlin/dukat for details

package org.w3c.dom.mediacapture

import
kotlin.js.*
import org.khronos.webgl.*
import org.w3c.dom.*
import org.w3c.dom.events.*

/**
 * Exposes
 * the JavaScript [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream)
 * to Kotlin
 */

public
external open class MediaStream() : EventTarget, MediaProvider {
 constructor(stream: MediaStream)
 constructor(tracks: Array<MediaStreamTrack>)

 open val id:
String
 open val active: Boolean
 var onaddtrack: ((MediaStreamTrackEvent) -> dynamic)?
 var
onremovetrack: ((MediaStreamTrackEvent) -> dynamic)?
 fun getAudioTracks(): Array<MediaStreamTrack>
 fun
getVideoTracks(): Array<MediaStreamTrack>
 fun
getTracks(): Array<MediaStreamTrack>
 fun
getTrackById(trackId: String): MediaStreamTrack?
 fun
addTrack(track: MediaStreamTrack)
 fun
removeTrack(track: MediaStreamTrack)
 fun
clone(): MediaStream

/**
 * Exposes the JavaScript
 * [MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack)
 * to Kotlin
 */

public
external abstract class MediaStreamTrack : EventTarget {
 open val kind: String
 open val id: String
 open
val label: String
 open var enabled: Boolean
 open val muted: Boolean
 open var onmute: ((Event) ->
dynamic)?
 open var onunmute:
((Event) -> dynamic)?
 open val readyState: MediaStreamTrackState
 open var onended: ((Event) ->
dynamic)?
 open var onoverconstrained: ((Event) -> dynamic)?
 fun clone(): MediaStreamTrack
 fun
stop()
 fun
getCapabilities(): MediaTrackCapabilities
 fun
getConstraints(): MediaTrackConstraints
 fun
getSettings(): MediaTrackSettings
 fun
applyConstraints(constraints: MediaTrackConstraints =
definedExternally): Promise<Unit>

/**
 * Exposes the JavaScript
 * [MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstrain
ts)
 * to Kotlin
 */

public
external interface MediaTrackSupportedConstraints {
 var width: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var height: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var aspectRatio: Boolean? /* = true */
 /*
 get()
= definedExternally
 set(value) = definedExternally
 var frameRate: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var facingMode: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var resizeMode: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var volume: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var sampleRate: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var sampleSize: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var echoCancellation: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var autoGainControl: Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var noiseSuppression:
Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var latency:
Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var channelCount:
Boolean? /* = true */
 /*
 get() = definedExternally
 set(value) = definedExternally
 var deviceId:

```

```

Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var groupId:
Boolean? /* = true */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaTrackSupportedConstraints(width: Boolean? = true, height: Boolean? = true, aspectRatio: Boolean? = true,
frameRate: Boolean? = true, facingMode: Boolean? = true, resizeMode: Boolean? = true, volume: Boolean? = true,
sampleRate: Boolean? = true, sampleSize: Boolean? = true, echoCancellation: Boolean? = true,
autoGainControl: Boolean? = true, noiseSuppression: Boolean? = true, latency: Boolean? = true, channelCount:
Boolean? = true, deviceId: Boolean? = true, groupId: Boolean? = true): MediaTrackSupportedConstraints {\n val o
= js(\"({})\")\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n
o[\"frameRate\"] = frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n
o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n
o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] = autoGainControl\n
o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n
o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return o\n}\n\npublic external interface
MediaTrackCapabilities {\n var width: ULongRange?\n get() = definedExternally\n set(value) =
definedExternally\n var height:
 ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var aspectRatio:
 DoubleRange?\n get() = definedExternally\n set(value) = definedExternally\n var frameRate:
 DoubleRange?\n get() = definedExternally\n set(value) = definedExternally\n var facingMode:
 Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n var resizeMode:
 Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n var volume:
 DoubleRange?\n get() = definedExternally\n set(value) = definedExternally\n var sampleRate:
 ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var sampleSize:
 ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var echoCancellation:
 Array<Boolean>?\n get() = definedExternally\n set(value) = definedExternally\n var autoGainControl:
 Array<Boolean>?\n
 get() = definedExternally\n set(value) = definedExternally\n var noiseSuppression: Array<Boolean>?\n
 get() = definedExternally\n set(value) = definedExternally\n var latency: DoubleRange?\n get() =
definedExternally\n set(value) = definedExternally\n var channelCount: ULongRange?\n get() =
definedExternally\n set(value) = definedExternally\n var deviceId: String?\n get() = definedExternally\n
 set(value) = definedExternally\n var groupId: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackCapabilities(width:
ULongRange? = undefined, height: ULongRange? = undefined, aspectRatio: DoubleRange? = undefined,
frameRate: DoubleRange? = undefined, facingMode: Array<String>? = undefined, resizeMode: Array<String>? =
undefined, volume: DoubleRange?
= undefined, sampleRate: ULongRange? = undefined, sampleSize: ULongRange? = undefined, echoCancellation:
Array<Boolean>? = undefined, autoGainControl: Array<Boolean>? = undefined, noiseSuppression:
Array<Boolean>? = undefined, latency: DoubleRange? = undefined, channelCount: ULongRange? = undefined,
deviceId: String? = undefined, groupId: String? = undefined): MediaTrackCapabilities {\n val o = js(\"({})\")\n
o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] =
frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n
o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] =
echoCancellation\n o[\"autoGainControl\"] = autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n
o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] =
groupId\n return o\n}\n\n**\n
```

\* Exposes the JavaScript

```
[MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin\n*\npublic external interface MediaTrackConstraints : MediaTrackConstraintSet {\n var advanced:\n Array<MediaTrackConstraintSet>?\n get() = definedExternally\n set(value) =\n definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",\n \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraints(advanced:\n Array<MediaTrackConstraintSet>? = undefined, width: dynamic = undefined, height: dynamic = undefined,\n aspectRatio: dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined,\n resizeMode: dynamic = undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize:\n dynamic = undefined, echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined,\n noiseSuppression: dynamic = undefined, latency: dynamic = undefined, channelCount: dynamic = undefined,\n deviceId: dynamic = undefined, groupId: dynamic = undefined): MediaTrackConstraints {\n val o = js(\"({})\")\n o[\"advanced\"] = advanced\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] =\n aspectRatio\n o[\"frameRate\"] = frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] =\n resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] = autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return o\n}\n\npublic external interface\n MediaTrackConstraintSet {\n var width: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var height: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var aspectRatio:\n dynamic\n get() = definedExternally\n set(value) = definedExternally\n var frameRate: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: dynamic\n get() =\n definedExternally\n set(value) = definedExternally\n var resizeMode: dynamic\n get() =\n definedExternally\n set(value) = definedExternally\n var volume: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var sampleRate: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var sampleSize: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var echoCancellation: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var autoGainControl: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var noiseSuppression: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var latency: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var channelCount: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n var deviceId: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var groupId: dynamic\n get() = definedExternally\n set(value) =\n definedExternally\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",\n \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraintSet(width:\n dynamic = undefined, height: dynamic = undefined, aspectRatio: dynamic = undefined, frameRate: dynamic =\n undefined, facingMode: dynamic = undefined, resizeMode: dynamic = undefined, volume: dynamic = undefined,\n sampleRate: dynamic = undefined, sampleSize: dynamic = undefined, echoCancellation: dynamic = undefined,\n autoGainControl: dynamic = undefined, noiseSuppression: dynamic = undefined, latency: dynamic\n = undefined, channelCount: dynamic = undefined, deviceId: dynamic = undefined, groupId: dynamic = undefined):\n MediaTrackConstraintSet {\n val o = js(\"({})\")\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] = frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] =\n autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return\n o\n }\n}\n\n/**\n * Exposes the JavaScript
```

```

[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin\n *\npublic
external interface MediaTrackSettings {\n var width: Int?\n get() = definedExternally\n
 set(value) = definedExternally\n var height: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var aspectRatio: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var frameRate: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var facingMode: String?\n get() = definedExternally\n set(value) =
definedExternally\n var resizeMode: String?\n get() = definedExternally\n set(value) =
definedExternally\n var volume: Double?\n get() = definedExternally\n set(value) = definedExternally\n
var sampleRate: Int?\n get() = definedExternally\n set(value) = definedExternally\n var sampleSize:
Int?\n get() = definedExternally\n set(value) = definedExternally\n var echoCancellation: Boolean?\n
get() = definedExternally\n set(value) = definedExternally\n var autoGainControl: Boolean?\n
 get() = definedExternally\n set(value) = definedExternally\n var noiseSuppression: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var latency: Double?\n get() = definedExternally\n
 set(value) = definedExternally\n var channelCount: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var deviceId: String?\n get() = definedExternally\n set(value) = definedExternally\n
var groupId: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackSettings(width: Int? =
undefined, height: Int? = undefined, aspectRatio: Double? = undefined, frameRate: Double? = undefined,
facingMode: String? = undefined, resizeMode: String? = undefined, volume: Double? = undefined, sampleRate: Int?
= undefined, sampleSize: Int? = undefined, echoCancellation:
Boolean? = undefined, autoGainControl: Boolean? = undefined, noiseSuppression: Boolean? = undefined, latency:
Double? = undefined, channelCount: Int? = undefined, deviceId: String? = undefined, groupId: String? =
undefined): MediaTrackSettings {\n val o = js(\"({})\")\n o[\"width\"] = width\n o[\"height\"] = height\n
o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] = frameRate\n o[\"facingMode\"] = facingMode\n
o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n
o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] =
autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n
o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return
o}\n}\n\n/*\n * Exposes the JavaScript
[MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin\n
*\npublic external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) :
Event {\n open val track: MediaStreamTrack\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
 }\n}\n\npublic external interface MediaStreamTrackEventInit : EventInit {\n var track:
MediaStreamTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamTrackEventInit(track:
MediaStreamTrack?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MediaStreamTrackEventInit {\n val o = js(\"({})\")\n o[\"track\"] = track\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o}\n}\n\npublic external open class
OverconstrainedErrorEvent(type: String, eventInitDict: OverconstrainedErrorEventInit) : Event {\n
 open val error: dynamic\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
 }\n}\n\npublic external interface OverconstrainedErrorEventInit : EventInit {\n var error: dynamic /* = null */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun OverconstrainedErrorEventInit(error:
dynamic = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
OverconstrainedErrorEventInit {\n val o = js(\"({})\")\n o[\"error\"] = error\n o[\"bubbles\"] = bubbles\n

```

```

o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin\n */\npublic external
abstract class MediaDevices : EventTarget
{\n open var ondevicechange: ((Event) -> dynamic)?\n fun enumerateDevices():
Promise<Array<MediaDeviceInfo>>\n fun getSupportedConstraints(): MediaTrackSupportedConstraints\n fun
getUserMedia(constraints: MediaStreamConstraints = definedExternally): Promise<MediaStream>\n}\n\n/**\n *
Exposes the JavaScript [MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to
Kotlin\n */\npublic external abstract class MediaDeviceInfo {\n open val deviceId: String\n open val kind:
MediaDeviceKind\n open val label: String\n open val groupId: String\n fun toJSON(): dynamic\n}\n\npublic
external abstract class InputDeviceInfo : MediaDeviceInfo {\n fun getCapabilities():
MediaTrackCapabilities\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to Kotlin\n
/\npublic external interface MediaStreamConstraints {\n var video: dynamic / = false */\n get()
= definedExternally\n set(value) = definedExternally\n var audio: dynamic /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamConstraints(video:
dynamic = false, audio: dynamic = false): MediaStreamConstraints {\n val o = js(\"({})\")\n o[\"video\"] =
video\n o[\"audio\"] = audio\n return o\n}\n\npublic external interface ConstrainablePattern {\n var
onoverconstrained: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
fun getCapabilities(): Capabilities\n fun getConstraints(): Constraints\n fun getSettings(): Settings\n fun
applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the
JavaScript [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n */\npublic
external interface
DoubleRange {\n var max: Double?\n get() = definedExternally\n set(value) = definedExternally\n var
min: Double?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleRange(max: Double? =
undefined, min: Double? = undefined): DoubleRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] =
min\n return o\n}\n\npublic external interface ConstrainDoubleRange : DoubleRange {\n var exact: Double?\n
get() = definedExternally\n set(value) = definedExternally\n var ideal: Double?\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainDoubleRange(exact: Double?
= undefined, ideal: Double? = undefined, max: Double? = undefined, min: Double? = undefined):
ConstrainDoubleRange
{\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n o[\"max\"] = max\n o[\"min\"] =
min\n return o\n}\n\npublic external interface ULongRange {\n var max: Int?\n get() = definedExternally\n
set(value) = definedExternally\n var min: Int?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange(max: Int? = undefined,
min: Int? = undefined): ULongRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] = min\n return
o\n}\n\npublic external interface ConstrainULongRange : ULongRange {\n var exact: Int?\n get() =
definedExternally\n set(value) = definedExternally\n var ideal: Int?\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline fun ConstrainULongRange(exact: Int? = undefined, ideal: Int? = undefined, max: Int? = undefined, min: Int?
= undefined): ConstrainULongRange {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n
o[\"max\"] = max\n o[\"min\"] = min\n return o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to

```

```

Kotlin\n *\npublic external interface ConstrainBooleanParameters {\n var exact: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var ideal: Boolean?\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainBooleanParameters(exact:
Boolean? = undefined, ideal: Boolean? = undefined): ConstrainBooleanParameters {\n val o = js("{}")\n
o["exact"] = exact\n o["ideal"]
= ideal\n return o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters)
to Kotlin\n *\npublic external interface ConstrainDOMStringParameters {\n var exact: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var ideal: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined):
ConstrainDOMStringParameters {\n val o = js("{}")\n o["exact"] = exact\n o["ideal"] = ideal\n return
o\n}\n\npublic external interface Capabilities\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Capabilities(): Capabilities {\n val o
= js("{}")\n return o\n}\n\npublic
external interface Settings\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Settings(): Settings {\n val o =
js("{}")\n return o\n}\n\npublic external interface ConstraintSet\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstraintSet(): ConstraintSet {\n
val o = js("{}")\n return o\n}\n\npublic external interface Constraints : ConstraintSet {\n var advanced:
Array<ConstraintSet>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Constraints(advanced:
Array<ConstraintSet>? = undefined): Constraints {\n val o = js("{}")\n o["advanced"] = advanced\n
return o\n}\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface
 MediaStreamTrackState {\n companion object\n}\n\npublic inline val MediaStreamTrackState.Companion.LIVE:
MediaStreamTrackState get() = "live".asDynamic().unsafeCast<MediaStreamTrackState>()\n\npublic inline val
MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() =
"ended".asDynamic().unsafeCast<MediaStreamTrackState>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoFacingModeEnum {\n companion object\n}\n\npublic inline val
VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum get() =
"user".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.ENVIRONMENT: VideoFacingModeEnum get() =
"environment".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() =
"left".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() =
"right".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoResizeModeEnum {\n companion object\n}\n\npublic inline val
VideoResizeModeEnum.Companion.NONE: VideoResizeModeEnum get() =
"none".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\npublic inline val
VideoResizeModeEnum.Companion.CROP_AND_SCALE: VideoResizeModeEnum get() = "crop-and-

```

```

scale\".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaDeviceKind {\n companion object\n}\n\npublic inline val
MediaDeviceKind.Companion.AUDIOINPUT: MediaDeviceKind get() =
\"audioinput\".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.AUDIOOUTPUT:
 MediaDeviceKind get() = \"audiooutput\".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.VIDEOINPUT: MediaDeviceKind get() =
\"videoinput\".asDynamic().unsafeCast<MediaDeviceKind>())\"/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediasource\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes
the JavaScript [MediaSource](https://developer.mozilla.org/en/docs/Web/API/MediaSource) to Kotlin\n */\n\npublic
external open class MediaSource : EventTarget, MediaProvider {\n open val sourceBuffers: SourceBufferList\n
open val activeSourceBuffers: SourceBufferList\n open val readyState: ReadyState\n
 var duration: Double\n var onsourceopen: ((Event) -> dynamic)?\n var onsourceended: ((Event) ->
dynamic)?\n var onsourceclose: ((Event) -> dynamic)?\n fun addSourceBuffer(type: String): SourceBuffer\n
fun removeSourceBuffer(sourceBuffer: SourceBuffer)\n fun endOfStream(error: EndOfStreamError =
definedExternally)\n fun setLiveSeekableRange(start: Double, end: Double)\n fun clearLiveSeekableRange()\n\n
companion object {\n fun isTypeSupported(type: String): Boolean\n }\n}\n\n/**\n * Exposes the JavaScript
[SourceBuffer](https://developer.mozilla.org/en/docs/Web/API/SourceBuffer) to Kotlin\n */\n\npublic external
abstract class SourceBuffer : EventTarget {\n open var mode: AppendMode\n open val updating: Boolean\n
open val buffered: TimeRanges\n open var timestampOffset: Double\n open val audioTracks: AudioTrackList\n
open val videoTracks: VideoTrackList\n open val textTracks: TextTrackList\n open var appendWindowStart:
Double\n
 open var appendWindowEnd: Double\n open var onupdatestart: ((Event) -> dynamic)?\n open var onupdate:
((Event) -> dynamic)?\n open var onupdateend: ((Event) -> dynamic)?\n open var onerror: ((Event) ->
dynamic)?\n open var onabort: ((Event) -> dynamic)?\n fun appendBuffer(data: dynamic)\n fun abort()\n fun
remove(start: Double, end: Double)\n}\n\n/**\n * Exposes the JavaScript
[SourceBufferList](https://developer.mozilla.org/en/docs/Web/API/SourceBufferList) to Kotlin\n */\n\npublic
external abstract class SourceBufferList : EventTarget {\n open val length: Int\n open var onaddsourcebuffer:
((Event) -> dynamic)?\n open var onremovesourcebuffer: ((Event) ->
dynamic)?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SourceBufferList.get(index:
Int): SourceBuffer? = asDynamic()[index]\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface ReadyState {\n companion object\n}\n\npublic inline val ReadyState.Companion.CLOSED:
ReadyState get() = \"closed\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val
ReadyState.Companion.OPEN: ReadyState get() = \"open\".asDynamic().unsafeCast<ReadyState>()\n\npublic
inline val ReadyState.Companion.ENDED: ReadyState get() =
\"ended\".asDynamic().unsafeCast<ReadyState>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface EndOfStreamError {\n companion object\n}\n\npublic inline val
EndOfStreamError.Companion.NETWORK: EndOfStreamError get() =
\"network\".asDynamic().unsafeCast<EndOfStreamError>()\n\npublic inline val
EndOfStreamError.Companion.DECODE: EndOfStreamError get() =
\"decode\".asDynamic().unsafeCast<EndOfStreamError>()\n\n/* please, don't implement this interface!

```



```

*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic
external interface AppendMode {\n companion object\n}\n\npublic inline val
AppendMode.Companion.SEGMENTS: AppendMode get() =
"\segments".asDynamic().unsafeCast<AppendMode>()\n\npublic inline val
AppendMode.Companion.SEQUENCE: AppendMode get() =
"\sequence".asDynamic().unsafeCast<AppendMode>()\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n*/\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.pointerevents\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface
PointerEventInit : MouseEventInit {\n var pointerId: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var width: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var pressure: Float? /* = 0f */\n get() = definedExternally\n set(value) =
definedExternally\n var tangentialPressure: Float? /* = 0f */\n get() = definedExternally\n set(value) =
definedExternally\n var tiltX: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n
 var tiltY: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var twist: Int? /* =
0 */\n get() = definedExternally\n set(value) = definedExternally\n var pointerType: String? /* = "" */\n
 get() = definedExternally\n set(value) = definedExternally\n var isPrimary: Boolean? /* = false */\n
 get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun PointerEventInit(pointerId: Int? = 0, width: Double? = 1.0, height: Double? = 1.0, pressure: Float? = 0f,
tangentialPressure: Float? = 0f, tiltX: Int? = 0, tiltY: Int? = 0, twist: Int? = 0, pointerType: String? = "", isPrimary:
Boolean? = false, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0,
buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false,
modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean?
= false, cancelable: Boolean? = false, composed: Boolean? = false): PointerEventInit {\n val o = js("{}")\n
o["pointerId"] = pointerId\n o["width"] = width\n o["height"] = height\n o["pressure"] = pressure\n
o["tangentialPressure"] = tangentialPressure\n o["tiltX"] = tiltX\n o["tiltY"] = tiltY\n o["twist"] = twist\n
o["pointerType"] = pointerType\n o["isPrimary"] = isPrimary\n o["screenX"] = screenX\n o["screenY"]
= screenY\n o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n o["buttons"] =
buttons\n o["relatedTarget"] = relatedTarget\n o["region"] = region\n o["ctrlKey"] = ctrlKey\n
o["shiftKey"] = shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"] =
modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"]
= modifierHyper\n o["modifierNumLock"] = modifierNumLock\n o["modifierScrollLock"] =
modifierScrollLock\n o["modifierSuper"] = modifierSuper\n o["modifierSymbol"] = modifierSymbol\n
o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] = view\n o["detail"] = detail\n
o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript
[PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin\n *\n*/\n\npublic external open
class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {\n open val
pointerId: Int\n open val width: Double\n open val height: Double\n open val pressure: Float\n open val
tangentialPressure: Float\n open val tiltX: Int\n open val tiltY: Int\n open val twist: Int\n open val

```

```

pointerType: String\n open val isPrimary: Boolean\n\n companion object {\n
 val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
 BUBBLING_PHASE: Short\n } \n}"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
 Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
 license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n// See
 github.com/kotlin/dukat for details\n\npackage org.w3c.dom.svg\n\nimport kotlin.js.*\nimport
 org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.css.*\n\n/**\n * Exposes the JavaScript
 [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin\n */\n\npublic external
 abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {\n
 open val dataset: DOMStringMap\n open val ownerSVGElement: SVGSVGElement?\n open val
 viewportElement: SVGElement?\n open var tabIndex: Int\n fun focus()\n fun blur()\n\n
 companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
 TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
 Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
 COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n}\n\npublic external interface
 SVGBoundingBoxOptions {\n
 var fill: Boolean? /* = true */\n get() = definedExternally\n set(value) =
 definedExternally\n var stroke: Boolean? /* = false */\n get()
 = definedExternally\n set(value) = definedExternally\n var markers: Boolean? /* = false */\n get() =
 definedExternally\n set(value) = definedExternally\n var clipped: Boolean? /* = false */\n get() =
 definedExternally\n set(value) = definedExternally\n } \n\n@Suppress("INVISIBLE_REFERENCE",
 "INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun SVGBoundingBoxOptions(fill:
 Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean? = false):
 SVGBoundingBoxOptions {\n
 val o = js("{}")\n o["fill"] = fill\n o["stroke"] = stroke\n o["markers"]
 = markers\n o["clipped"] = clipped\n return o\n } \n\n/**\n * Exposes the JavaScript
 [SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin\n
 */\n\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n
 open val transform:
 SVGAnimatedTransformList\n fun getBBox(options:
 SVGBoundingBoxOptions = definedExternally): DOMRect\n fun
 getCTM(): DOMMatrix?\n fun
 getScreenCTM(): DOMMatrix?\n\n companion object {\n
 val ELEMENT_NODE: Short\n val
 ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
 ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
 PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n}\n\n/**\n * Exposes the JavaScript
 [SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement)
 to Kotlin\n */\n\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {\n
 open val
 pathLength: SVGAnimatedNumber\n fun isPointInFill(point: DOMPoint): Boolean\n fun isPointInStroke(point:
 DOMPoint): Boolean\n fun getTotalLength(): Float\n fun getPointAtLength(distance: Float): DOMPoint\n\n
 companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val

```

```

TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin \n * \n public external
abstract class SVGNumber { \n open var value: Float \n} \n \n /** \n * Exposes the JavaScript
[SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin \n * \n public external abstract
class SVGLength { \n open val unitType: Short \n open var value: Float \n open var valueInSpecifiedUnits:
Float \n open var valueAsString: String \n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:
Float) \n fun convertToSpecifiedUnits(unitType: Short) \n \n companion object { \n val
SVG_LENGTHTYPE_UNKNOWN: Short \n val SVG_LENGTHTYPE_NUMBER: Short \n val
SVG_LENGTHTYPE_PERCENTAGE: Short \n val SVG_LENGTHTYPE_EMS: Short \n val
SVG_LENGTHTYPE_EXS: Short \n val SVG_LENGTHTYPE_PX:
Short \n val SVG_LENGTHTYPE_CM: Short \n val SVG_LENGTHTYPE_MM: Short \n val
SVG_LENGTHTYPE_IN: Short \n val SVG_LENGTHTYPE_PT: Short \n val SVG_LENGTHTYPE_PC:
Short \n } \n} \n \n /** \n * Exposes the JavaScript
[SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin \n * \n public external abstract
class SVGAngle { \n open val unitType: Short \n open var value: Float \n open var valueInSpecifiedUnits:
Float \n open var valueAsString: String \n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:
Float) \n fun convertToSpecifiedUnits(unitType: Short) \n \n companion object { \n val
SVG_ANGLETYPE_UNKNOWN: Short \n val SVG_ANGLETYPE_UNSPECIFIED: Short \n val
SVG_ANGLETYPE_DEG: Short \n val SVG_ANGLETYPE_RAD: Short \n val
SVG_ANGLETYPE_GRAD: Short \n } \n} \n \n public external abstract class SVGNameList { \n open val length:
Int \n open val numberOfItems: Int \n fun clear() \n
 fun initialize(newItem: dynamic): dynamic \n fun insertItemBefore(newItem: dynamic, index: Int): dynamic \n
 fun replaceItem(newItem: dynamic, index: Int): dynamic \n fun removeItem(index: Int): dynamic \n fun
appendItem(newItem: dynamic): dynamic \n fun getItem(index: Int):
dynamic \n} \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGNameList.get(index: Int):
dynamic = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGNameList.set(index: Int,
newItem: dynamic) { asDynamic()[index] = newItem } \n \n /** \n * Exposes the JavaScript
[SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin \n * \n public external
abstract class SVGNumberList { \n open val length: Int \n open val numberOfItems: Int \n fun clear() \n fun
initialize(newItem: SVGNumber): SVGNumber \n fun insertItemBefore(newItem:
SVGNumber, index: Int): SVGNumber \n fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber \n
 fun removeItem(index: Int): SVGNumber \n fun appendItem(newItem: SVGNumber): SVGNumber \n fun
getItem(index: Int): SVGNumber \n} \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGNumberList.get(index:
Int): SVGNumber? = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGNumberList.set(index:
Int, newItem: SVGNumber) { asDynamic()[index] = newItem } \n \n /** \n * Exposes the JavaScript
[SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin \n * \n public external
abstract class SVGLengthList { \n open val length: Int \n open val numberOfItems: Int \n fun clear() \n fun

```

```

initialize(newItem: SVGLength): SVGLength\n fun insertItemBefore(newItem: SVGLength, index: Int):
SVGLength\n
fun replaceItem(newItem: SVGLength, index: Int): SVGLength\n fun removeItem(index: Int): SVGLength\n
fun appendItem(newItem: SVGLength): SVGLength\n fun getItem(index: Int):
SVGLength\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.get(index:
Int): SVGLength? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.set(index: Int,
newItem: SVGLength) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin\n
*/\npublic external abstract class SVGAnimatedBoolean {\n open var baseVal: Boolean\n open val animVal:
Boolean\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to
Kotlin\n */\npublic external abstract
class SVGAnimatedEnumeration {\n open var baseVal: Short\n open val animVal: Short\n}\n\n/**\n * Exposes
the JavaScript [SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to
Kotlin\n */\npublic external abstract class SVGAnimatedInteger {\n open var baseVal: Int\n open val animVal:
Int\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin\n
*/\npublic external abstract class SVGAnimatedNumber {\n open var baseVal: Float\n open val animVal:
Float\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin\n
*/\npublic external abstract class SVGAnimatedLength {\n open val baseVal: SVGLength\n open val animVal:
SVGLength\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin\n */\npublic
external abstract
class SVGAnimatedAngle {\n open val baseVal: SVGAngle\n open val animVal: SVGAngle\n}\n\n/**\n *
Exposes the JavaScript
[SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin\n */\npublic
external abstract class SVGAnimatedString {\n open var baseVal: String\n open val animVal: String\n}\n\n/**\n
* Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect)
to Kotlin\n */\npublic external abstract class SVGAnimatedRect {\n open val baseVal: DOMRect\n open val
animVal: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin\n
*/\npublic external abstract class SVGAnimatedNumberList {\n open val baseVal: SVGNumberList\n open val
animVal: SVGNumberList\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to
Kotlin\n */\npublic external abstract class SVGAnimatedLengthList {\n open val baseVal: SVGLengthList\n
open val animVal: SVGLengthList\n}\n\n/**\n * Exposes the JavaScript
[SVGStringList](https://developer.mozilla.org/en/docs/Web/API/SVGStringList) to Kotlin\n */\npublic external
abstract class SVGStringList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun
initialize(newItem: String): String\n fun insertItemBefore(newItem: String, index: Int): String\n fun
replaceItem(newItem: String, index: Int): String\n fun removeItem(index: Int): String\n fun
appendItem(newItem: String): String\n fun getItem(index: Int):
String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.get(index:
Int): String? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.set(index:

```

```

Int, newItem: String) { asDynamic()[index] = newItem } \n \n ** \n * Exposes the JavaScript
[SVGUnitTypes](https://developer.mozilla.org/en/docs/Web/API/SVGUnitTypes) to Kotlin \n
*\n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n public external interface SVGUnitTypes
{\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short \n val
SVG_UNIT_TYPE_USERSPACEONUSE: Short \n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short \n } \n } \n \n ** \n * Exposes the JavaScript
[SVGTTests](https://developer.mozilla.org/en/docs/Web/API/SVGTTests) to Kotlin \n * \n public external interface
SVGTTests {\n val requiredExtensions: SVGStringList \n val systemLanguage: SVGStringList \n } \n \n public
external interface SVGFitToViewBox {\n val viewBox: SVGAnimatedRect \n val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio \n } \n \n ** \n * Exposes the JavaScript
[SVGZoomAndPan](https://developer.mozilla.org/en/docs/Web/API/SVGZoomAndPan) to Kotlin \n
*\n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n public
external interface SVGZoomAndPan {\n var zoomAndPan: Short \n \n companion object {\n val
SVG_ZOOMANDPAN_UNKNOWN: Short \n val SVG_ZOOMANDPAN_DISABLE: Short \n val
SVG_ZOOMANDPAN_MAGNIFY: Short \n } \n } \n \n ** \n * Exposes the JavaScript
[SVGURIReference](https://developer.mozilla.org/en/docs/Web/API/SVGURIReference) to Kotlin \n * \n public
external interface SVGURIReference {\n val href: SVGAnimatedString \n } \n \n ** \n * Exposes the JavaScript
[SVGSVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGSVGElement) to Kotlin \n * \n public
external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan,
WindowEventHandlers {\n open val x: SVGAnimatedLength \n open val y: SVGAnimatedLength \n open val
width: SVGAnimatedLength \n open val height: SVGAnimatedLength \n open var currentScale: Float \n open
val currentTranslate: DOMPointReadOnly \n fun getIntersectionList(rect: DOMRectReadOnly, referenceElement:
SVGElement?): NodeList \n fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?):
NodeList \n fun checkIntersection(element: SVGElement, rect: DOMRectReadOnly): Boolean \n fun
checkEnclosure(element: SVGElement, rect: DOMRectReadOnly): Boolean \n fun deselectAll() \n fun
createSVGNumber(): SVGNumber \n fun createSVGLength(): SVGLength \n fun createSVGAngle():
SVGAngle \n fun createSVGPoint(): DOMPoint \n fun createSVGMatrix(): DOMMatrix \n fun
createSVGRect(): DOMRect \n fun createSVGTransform(): SVGTransform \n fun
createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform \n fun
getElementById(elementId: String): Element \n fun suspendRedraw(maxWaitMilliseconds: Int): Int \n fun
unsuspendRedraw(suspendHandleID: Int) \n fun unsuspendRedrawAll() \n fun forceRedraw() \n \n companion
object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short \n val SVG_ZOOMANDPAN_DISABLE:
Short \n val SVG_ZOOMANDPAN_MAGNIFY:
Short \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n
 val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n } \n \n ** \n * Exposes the JavaScript
[SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin \n * \n public external
abstract class SVGGElement : SVGGraphicsElement {\n companion object {\n val ELEMENT_NODE:
Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:
Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val

```

DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val

DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n  
val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n  
val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n }n}n\n\npublic external abstract class  
SVGUnknownElement : SVGGraphicsElement {\n companion object {\n val ELEMENT\_NODE: Short\n  
val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val CDATA\_SECTION\_NODE: Short\n  
val ENTITY\_REFERENCE\_NODE:  
Short\n val ENTITY\_NODE: Short\n val PROCESSING\_INSTRUCTION\_NODE: Short\n val  
COMMENT\_NODE: Short\n val DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n  
val DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val  
DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n  
val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n  
val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n }n}n\n\n/\*\*\n \* Exposes the JavaScript  
[SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin\n \*/\npublic  
external abstract class SVGDefsElement : SVGGraphicsElement {\n companion object {\n val  
ELEMENT\_NODE: Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val  
CDATA\_SECTION\_NODE: Short\n val ENTITY\_REFERENCE\_NODE: Short\n  
val ENTITY\_NODE: Short\n val PROCESSING\_INSTRUCTION\_NODE: Short\n val  
COMMENT\_NODE: Short\n val DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n  
val DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val  
DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n  
val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n  
val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n }n}n\n\n/\*\*\n \* Exposes the JavaScript  
[SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin\n \*/\npublic  
external abstract class SVGDescElement : SVGElement {\n companion object {\n val ELEMENT\_NODE:  
Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val CDATA\_SECTION\_NODE:  
Short\n val ENTITY\_REFERENCE\_NODE: Short\n val ENTITY\_NODE: Short\n  
val PROCESSING\_INSTRUCTION\_NODE: Short\n val COMMENT\_NODE: Short\n val  
DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n val  
DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val  
DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n  
val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n  
val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n }n}n\n\n/\*\*\n \* Exposes the JavaScript  
[SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin\n \*/\npublic external abstract class SVGMetadataElement : SVGElement {\n companion object {\n val  
ELEMENT\_NODE: Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val  
CDATA\_SECTION\_NODE: Short\n val ENTITY\_REFERENCE\_NODE: Short\n val ENTITY\_NODE:  
Short\n val  
PROCESSING\_INSTRUCTION\_NODE: Short\n val COMMENT\_NODE: Short\n val  
DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n val  
DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val  
DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n  
val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n  
val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val



```

correspondingUseElement: SVGUseElement? \n get() = definedExternally \n} \n \n public external open class
ShadowAnimation(source: dynamic, newTarget: dynamic) { \n open val sourceAnimation: dynamic \n} \n \n /** \n *
Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement)
to Kotlin \n * \n public external abstract class SVGSwitchElement : SVGGraphicsElement { \n companion object
{ \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n
val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n public external interface
GetSVGDocument { \n fun getSVGDocument():
Document \n} \n \n /** \n * Exposes the JavaScript
[SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin \n * \n public
external abstract class SVGStyleElement : SVGElement, LinkStyle { \n open var type: String \n open var media:
String \n open var title: String \n \n companion object { \n val ELEMENT_NODE: Short \n val
ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val
ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short \n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n * Exposes
the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin \n * \n
public external abstract class SVGTransform { \n open val type: Short \n open val matrix: DOMMatrix \n
open val angle: Float \n fun setMatrix(matrix: DOMMatrixReadOnly) \n fun setTranslate(tx: Float, ty: Float) \n
fun setScale(sx: Float, sy: Float) \n fun setRotate(angle: Float, cx: Float, cy: Float) \n fun setSkewX(angle:
Float) \n fun setSkewY(angle: Float) \n \n companion object { \n val SVG_TRANSFORM_UNKNOWN:
Short \n val SVG_TRANSFORM_MATRIX: Short \n val SVG_TRANSFORM_TRANSLATE: Short \n
 val SVG_TRANSFORM_SCALE: Short \n val SVG_TRANSFORM_ROTATE: Short \n val
SVG_TRANSFORM_SKEWX: Short \n val SVG_TRANSFORM_SKEWY: Short \n } \n} \n \n /** \n * Exposes
the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList)
to Kotlin \n * \n public external abstract class SVGTransformList { \n open val length: Int \n open val
numberOfItems: Int \n fun clear() \n fun initialize(newItem: SVGTransform): SVGTransform \n fun
insertItemBefore(newItem: SVGTransform, index: Int): SVGTransform \n fun replaceItem(newItem:
SVGTransform, index: Int): SVGTransform \n fun removeItem(index: Int): SVGTransform \n fun
appendItem(newItem: SVGTransform): SVGTransform \n fun createSVGTransformFromMatrix(matrix:
DOMMatrixReadOnly): SVGTransform \n fun consolidate(): SVGTransform? \n fun getItem(index: Int):
SVGTransform \n} \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGTransformList.get(index:
Int): SVGTransform? = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline operator fun SVGTransformList.set(index:
Int, newItem: SVGTransform)

```



```

{ asDynamic()[index] = newItem } \n\n/** \n * Exposes the JavaScript
[SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to
Kotlin \n * \npublic external abstract class SVGAnimatedTransformList { \n open val baseVal:
SVGTransformList \n open val animVal: SVGTransformList \n} \n\n/** \n * Exposes the JavaScript
[SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin \n
* \npublic external abstract class SVGPreserveAspectRatio { \n open var align: Short \n open var meetOrSlice:
Short \n \n companion object { \n val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short \n val
SVG_PRESERVEASPECTRATIO_NONE: Short \n val SVG_PRESERVEASPECTRATIO_XMINYMIN:
Short \n val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short \n val
SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short \n val
SVG_PRESERVEASPECTRATIO_XMINYMID: Short \n val
SVG_PRESERVEASPECTRATIO_XMIDYMID: Short \n val
SVG_PRESERVEASPECTRATIO_XMAXYMID:
Short \n val SVG_PRESERVEASPECTRATIO_XMINYMAX: Short \n val
SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short \n val
SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short \n val SVG_MEETORSLICE_UNKNOWN: Short \n
 val SVG_MEETORSLICE_MEET: Short \n val SVG_MEETORSLICE_SLICE: Short \n } \n} \n\n/** \n *
Exposes the JavaScript
[SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspect
Ratio) to Kotlin \n * \npublic external abstract class SVGAnimatedPreserveAspectRatio { \n open val baseVal:
SVGPreserveAspectRatio \n open val animVal: SVGPreserveAspectRatio \n} \n\n/** \n * Exposes the JavaScript
[SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin \n * \npublic
external abstract class SVGPathElement : SVGGeometryElement { \n companion object { \n val
ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE:
Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n\n/** \n * Exposes the JavaScript
[SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin \n * \npublic
external abstract class SVGRectElement : SVGGeometryElement { \n open val x: SVGAnimatedLength \n open
val y: SVGAnimatedLength \n open val width: SVGAnimatedLength \n open val height: SVGAnimatedLength \n
 open val rx: SVGAnimatedLength \n
 open val ry: SVGAnimatedLength \n \n companion object { \n val ELEMENT_NODE: Short \n val
ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val
ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val
PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n\n/** \n * Exposes the JavaScript
[SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin \n * \npublic

```

```

external abstract class SVGCircleElement
: SVGGeometryElement {
 open val cx: SVGAnimatedLength
 open val cy: SVGAnimatedLength
 open val r: SVGAnimatedLength
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

/** Exposes the JavaScript [SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin */
public external abstract class SVGEllipseElement : SVGGeometryElement {
 open val cx: SVGAnimatedLength
 open val cy: SVGAnimatedLength
 open val rx: SVGAnimatedLength
 open val ry: SVGAnimatedLength
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

/** Exposes the JavaScript [SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin */
public external abstract class SVGLineElement : SVGGeometryElement {
 open val x1: SVGAnimatedLength
 open val y1: SVGAnimatedLength
 open val x2: SVGAnimatedLength
 open val y2: SVGAnimatedLength
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

/** Exposes the JavaScript [SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin */
public external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 }
}

```

```

Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin\n
*/\npublic external interface SVGAnimatedPoints {\n val points: SVGPointList\n val animatedPoints:
SVGPointList\n}\n\npublic external abstract class SVGPointList {\n open val length: Int\n open val
numberOfItems: Int\n fun clear()\n fun initialize(newItem: DOMPoint): DOMPoint\n fun
insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint\n fun replaceItem(newItem: DOMPoint, index:
Int): DOMPoint\n fun removeItem(index: Int): DOMPoint\n fun appendItem(newItem: DOMPoint):
DOMPoint\n fun getItem(index: Int): DOMPoint\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.get(index:
Int): DOMPoint? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.set(index: Int,
newItem: DOMPoint) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin\n
*/\npublic external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin\n
*/\npublic external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin\n
*/\npublic external abstract class SVGTextContentElement : SVGGraphicsElement {\n open val textLength:
SVGAnimatedLength\n open val lengthAdjust: SVGAnimatedEnumeration\n fun getNumberOfChars(): Int\n
fun getComputedTextLength(): Float\n fun getSubStringLength(charnum: Int, nchars: Int): Float\n fun
getStartPositionOfChar(charnum: Int): DOMPoint\n fun getEndPositionOfChar(charnum: Int): DOMPoint\n fun
getExtentOfChar(charnum: Int): DOMRect\n fun getRotationOfChar(charnum: Int): Float\n fun
getCharNumAtPosition(point: DOMPoint): Int\n fun selectSubString(charnum:
Int, nchars: Int)\n\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val
LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val

```



DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n } \n\n\n \* Exposes the JavaScript [SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin\n\n \* \npublic external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference {\n open val startOffset: SVGAnimatedLength\n open val method: SVGAnimatedEnumeration\n open val spacing: SVGAnimatedEnumeration\n\n companion object {\n val TEXTPATH\_METHODTYPE\_UNKNOWN: Short\n val TEXTPATH\_METHODTYPE\_ALIGN: Short\n val TEXTPATH\_METHODTYPE\_STRETCH: Short\n val TEXTPATH\_SPACINGTYPE\_UNKNOWN: Short\n val TEXTPATH\_SPACINGTYPE\_AUTO: Short\n val TEXTPATH\_SPACINGTYPE\_EXACT: Short\n val LENGTHADJUST\_UNKNOWN: Short\n val LENGTHADJUST\_SPACING: Short\n val LENGTHADJUST\_SPACINGANDGLYPHS: Short\n val ELEMENT\_NODE: Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val CDATA\_SECTION\_NODE: Short\n val ENTITY\_REFERENCE\_NODE: Short\n val ENTITY\_NODE: Short\n val PROCESSING\_INSTRUCTION\_NODE: Short\n val COMMENT\_NODE: Short\n val DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n val DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n } \n\n\n \* Exposes the JavaScript [SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin\n\n \* \npublic external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference, HTMLOrSVGImageElement {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val preserveAspectRatio: SVGAnimatedPreserveAspectRatio\n open var crossOrigin: String?\n\n companion object {\n val ELEMENT\_NODE: Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val CDATA\_SECTION\_NODE: Short\n val ENTITY\_REFERENCE\_NODE: Short\n val ENTITY\_NODE: Short\n val PROCESSING\_INSTRUCTION\_NODE: Short\n val COMMENT\_NODE: Short\n val DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n val DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n } \n\n\n \* Exposes the JavaScript [SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to Kotlin\n\n \* \npublic external abstract class SVGForeignObjectElement : SVGGraphicsElement {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n\n companion object {\n val ELEMENT\_NODE: Short\n val ATTRIBUTE\_NODE: Short\n val TEXT\_NODE: Short\n val CDATA\_SECTION\_NODE: Short\n val ENTITY\_REFERENCE\_NODE: Short\n val ENTITY\_NODE: Short\n val PROCESSING\_INSTRUCTION\_NODE: Short\n val COMMENT\_NODE: Short\n val DOCUMENT\_NODE: Short\n val DOCUMENT\_TYPE\_NODE: Short\n val DOCUMENT\_FRAGMENT\_NODE: Short\n val NOTATION\_NODE: Short\n val DOCUMENT\_POSITION\_DISCONNECTED: Short\n val DOCUMENT\_POSITION\_PRECEDING: Short\n val DOCUMENT\_POSITION\_FOLLOWING: Short\n val DOCUMENT\_POSITION\_CONTAINS: Short\n val DOCUMENT\_POSITION\_CONTAINED\_BY: Short\n val

DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC: Short\n } \n\n\npublic external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {\n open val refX: SVGAnimatedLength\n open val

```

refY: SVGAnimatedLength\n open val markerUnits: SVGAnimatedEnumeration\n open val markerWidth:
SVGAnimatedLength\n open val markerHeight: SVGAnimatedLength\n open val orientType:
SVGAnimatedEnumeration\n open val orientAngle: SVGAnimatedAngle\n open var orient: String\n fun
setOrientToAuto()\n
fun setOrientToAngle(angle: SVGAngle)\n\n companion object {\n val
SVG_MARKERUNITS_UNKNOWN: Short\n val SVG_MARKERUNITS_USERSPACEONUSE: Short\n
val SVG_MARKERUNITS_STROKEWIDTH: Short\n val SVG_MARKER_ORIENT_UNKNOWN: Short\n
val SVG_MARKER_ORIENT_AUTO: Short\n val SVG_MARKER_ORIENT_ANGLE: Short\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n/**\n * Exposes
the JavaScript [SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to
Kotlin\n */\npublic external abstract class SVGSolidcolorElement : SVGElement {\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to
Kotlin\n */\npublic external abstract class SVGGradientElement : SVGElement, SVGURIReference,
SVGUnitTypes {\n open val gradientUnits: SVGAnimatedEnumeration\n open val gradientTransform:
SVGAnimatedTransformList\n open val spreadMethod: SVGAnimatedEnumeration\n\n companion object {\n
val SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val
SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val
SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to
Kotlin\n */\npublic external abstract class SVGLinearGradientElement : SVGGradientElement {\n open val x1:
SVGAnimatedLength\n open val y1: SVGAnimatedLength\n open val x2: SVGAnimatedLength\n open val

```

```

y2: SVGAnimatedLength\n\n companion object {\n val
 SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val
 SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val
 SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
 SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val
 ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
 ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
 PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/*\n * Exposes the
JavaScript
[SVGRadialGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGRadialGradientElement) to
Kotlin\n *\npublic external abstract class SVGRadialGradientElement : SVGGradientElement {\n open val cx:
SVGAnimatedLength\n open val cy: SVGAnimatedLength\n open val r: SVGAnimatedLength\n open val fx:
SVGAnimatedLength\n open val fy: SVGAnimatedLength\n open val fr: SVGAnimatedLength\n\n companion
object {\n val SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD:
Short\n val SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT:
Short\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE:
Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshGradientElement : SVGGradientElement {\n companion object {\n val
SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val
SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val
SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshrowElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:

```

```

Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshpatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n
 val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin\n */\npublic
external abstract class SVGStopElement : SVGElement {\n open val offset: SVGAnimatedNumber\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE:
Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val
DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val
NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to
Kotlin\n */\npublic external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox,
SVGURIReference, SVGUnitTypes {\n open val patternUnits: SVGAnimatedEnumeration\n open val
patternContentUnits: SVGAnimatedEnumeration\n open val patternTransform: SVGAnimatedTransformList\n
 open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open
 val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n\n companion object {\n
 val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic
external abstract class SVGHatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n

```



```

Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGHatchpathElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin\n */\npublic
external abstract class SVGCursorElement : SVGElement, SVGURIReference {\n open val x:
SVGAnimatedLength\n open val y: SVGAnimatedLength\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin\n */\npublic
external abstract class SVGScriptElement : SVGElement, SVGURIReference, HTMLOrSVGScriptElement
{\n open var type: String\n open var crossOrigin: String?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGAEElement](https://developer.mozilla.org/en/docs/Web/API/SVGAEElement) to Kotlin\n */\npublic external
abstract
class SVGAEElement : SVGGraphicsElement, SVGURIReference {\n open val target: SVGAnimatedString\n
open val download: SVGAnimatedString\n open val rel: SVGAnimatedString\n open val relList:
SVGAnimatedString\n open val hreflang: SVGAnimatedString\n open val type: SVGAnimatedString\n\n

```

```

companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/** Exposes the JavaScript [SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin
 *
 * public external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {
 *
 * companion object {
 * val SVG_ZOOMANDPAN_UNKNOWN: Short
 * val SVG_ZOOMANDPAN_DISABLE: Short
 * val SVG_ZOOMANDPAN_MAGNIFY: Short
 * val ELEMENT_NODE: Short
 * val ATTRIBUTE_NODE: Short
 * val TEXT_NODE: Short
 * val CDATA_SECTION_NODE: Short
 * val ENTITY_REFERENCE_NODE: Short
 * val ENTITY_NODE: Short
 * val PROCESSING_INSTRUCTION_NODE: Short
 * val COMMENT_NODE: Short
 * val DOCUMENT_NODE: Short
 * val DOCUMENT_TYPE_NODE: Short
 * val DOCUMENT_FRAGMENT_NODE: Short
 * val NOTATION_NODE: Short
 * val DOCUMENT_POSITION_DISCONNECTED: Short
 * val DOCUMENT_POSITION_PRECEDING: Short
 * val DOCUMENT_POSITION_FOLLOWING: Short
 * val DOCUMENT_POSITION_CONTAINS: Short
 * val DOCUMENT_POSITION_CONTAINED_BY: Short
 * val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 * }
 *
 * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 *
 * NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
 * See github.com/kotlin/dukat for details
 *
 * package org.w3c.files
 *
 * import kotlin.js.*
 * import org.khronos.webgl.*
 * import org.w3c.dom.*
 * import org.w3c.dom.events.*
 * import org.w3c.xhr.*
 *
 * /** Exposes the JavaScript [Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin
 *
 * public external open class Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) : MediaProvider, ImageBitmapSource {
 * open val size: Number
 * open val type: String
 *
 * open val isClosed: Boolean
 * fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String = definedExternally): Blob
 * fun close()
 *
 * public external interface BlobPropertyBag {
 * var type: String? /* = "" */
 * get() = definedExternally
 * set(value) = definedExternally
 *
 * @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
 * @kotlin.internal.InlineOnly
 * public inline fun BlobPropertyBag(type: String? = ""): BlobPropertyBag {
 * val o = js("{}")
 * o["type"] = type
 * return o
 *
 * /** Exposes the JavaScript [File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin
 *
 * public external open class File(fileBits: Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {
 * open val name: String
 * open val lastModified: Int
 *
 * public external interface FilePropertyBag : BlobPropertyBag {
 * var lastModified: Int?
 * get() = definedExternally
 *
 * set(value) = definedExternally
 *
 * @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
 * @kotlin.internal.InlineOnly
 * public inline fun FilePropertyBag(lastModified: Int? = undefined, type: String? = ""): FilePropertyBag {
 * val o = js("{}")
 * o["lastModified"] = lastModified
 * o["type"] = type
 * return o
 *
 * /** Exposes the JavaScript [FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin
 *
 * public external abstract class FileList : ItemArrayLike<File> {
 * override fun item(index: Int): File?
 *
 * @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
 * @kotlin.internal.InlineOnly
 * public inline operator fun FileList.get(index: Int): File?

```

```

= asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin\n */\npublic external open class
FileReader : EventTarget {\n open val readyState: Short\n open val result: dynamic\n open val error:
dynamic\n var onloadstart: ((ProgressEvent) -> dynamic)?\n var onprogress: ((ProgressEvent) -> dynamic)?\n
var onload: ((Event) -> dynamic)?\n var onabort: ((Event) -> dynamic)?\n var onerror: ((Event) -> dynamic)?\n
var onloadend: ((Event) -> dynamic)?\n fun readAsArrayBuffer(blob: Blob)\n fun readAsBinaryString(blob:
Blob)\n fun readAsText(blob: Blob, label: String = definedExternally)\n fun readAsDataURL(blob: Blob)\n
fun abort()\n\n companion object {\n val EMPTY: Short\n val LOADING: Short\n val DONE:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin\n */\npublic external
open class FileReaderSync {\n fun readAsArrayBuffer(blob: Blob): ArrayBuffer\n fun readAsBinaryString(blob:
Blob): String\n fun readAsText(blob: Blob, label: String = definedExternally): String\n fun
readAsDataURL(blob: Blob): String\n}", "/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS
AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.notifications\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.events.*\nimport
org.w3c.workers.*\n\n/**\n * Exposes the JavaScript
[Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin\n */\npublic external open
class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {\n var onclick:
((MouseEvent) -> dynamic)?\n var onerror: ((Event) -> dynamic)?\n open val title: String\n open val dir:
NotificationDirection\n open val lang: String\n open val body: String\n open val tag: String\n open val
image: String\n open val icon: String\n open val badge: String\n
open val sound: String\n open val vibrate: Array<out Int>\n open val timestamp: Number\n open val renotify:
Boolean\n open val silent: Boolean\n open val noscreen: Boolean\n open val requireInteraction: Boolean\n
open val sticky: Boolean\n open val data: Any?\n open val actions: Array<out NotificationAction>\n fun
close()\n\n companion object {\n val permission: NotificationPermission\n val maxActions: Int\n fun
requestPermission(deprecatedCallback: (NotificationPermission) -> Unit = definedExternally):
Promise<NotificationPermission>\n }\n}\n\npublic external interface NotificationOptions {\n var dir:
NotificationDirection? /* = NotificationDirection.AUTO */\n get() = definedExternally\n set(value) =
definedExternally\n var lang: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n var body: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n var tag: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n var image: String?\n get() = definedExternally\n set(value) = definedExternally\n
var icon: String?\n get() = definedExternally\n set(value) = definedExternally\n var badge: String?\n
get() = definedExternally\n set(value) = definedExternally\n var sound: String?\n get() =
definedExternally\n set(value) = definedExternally\n var vibrate: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n var timestamp: Number?\n get() = definedExternally\n set(value) =
definedExternally\n var renotify: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var silent: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var noscreen: Boolean? /* = false
/\n get() = definedExternally\n set(value) = definedExternally\n var requireInteraction: Boolean? / =
false */\n get() = definedExternally\n set(value) = definedExternally\n var sticky: Boolean? /* = false
/\n get() = definedExternally\n set(value) = definedExternally\n var data: Any? / = null */\n get() =
definedExternally\n set(value) = definedExternally\n var actions: Array<NotificationAction>? /* = arrayOf()
*/\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\npublic inline fun NotificationOptions(dir:

```

```

NotificationDirection? = NotificationDirection.AUTO, lang: String? = "", body: String? = "", tag: String? = "",
image: String? = undefined, icon: String? = undefined, badge: String? = undefined, sound: String? = undefined,
vibrate: dynamic = undefined, timestamp: Number? = undefined, renotify:
Boolean? = false, silent: Boolean? = false, noscreen: Boolean? = false, requireInteraction: Boolean? = false, sticky:
Boolean? = false, data: Any? = null, actions: Array<NotificationAction>? = arrayOf(): NotificationOptions {\n
val o = js("{}")\n o["dir"] = dir\n o["lang"] = lang\n o["body"] = body\n o["tag"] = tag\n
o["image"] = image\n o["icon"] = icon\n o["badge"] = badge\n o["sound"] = sound\n o["vibrate"] =
vibrate\n o["timestamp"] = timestamp\n o["renotify"] = renotify\n o["silent"] = silent\n o["noscreen"] =
noscreen\n o["requireInteraction"] = requireInteraction\n o["sticky"] = sticky\n o["data"] = data\n
o["actions"] = actions\n return o}\n\npublic external interface NotificationAction {\n var action: String?\n
var title: String?\n var icon: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationAction(action: String?,
title: String?, icon: String? = undefined): NotificationAction {\n val o = js("{}")\n o["action"] = action\n
o["title"] = title\n o["icon"] = icon\n return o}\n\npublic external interface GetNotificationOptions {\n var
tag: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun GetNotificationOptions(tag: String? =
 ""): GetNotificationOptions {\n val o = js("{}")\n o["tag"] = tag\n return o}\n\n/**\n * Exposes the
JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin\n
*\n\npublic external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) :
ExtendableEvent {\n open val notification: Notification\n
open val action: String\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n
val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
NotificationEventInit : ExtendableEventInit {\n var notification: Notification?\n var action: String? /* = "" */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationEventInit(notification:
Notification?, action: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): NotificationEventInit {\n val o = js("{}")\n o["notification"] = notification\n o["action"] =
action\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o}\n\n/* please, don't implement this interface!\n
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic
external interface NotificationPermission {\n companion object\n}\n\npublic inline val
NotificationPermission.Companion.DEFAULT: NotificationPermission get() =
"default".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val
NotificationPermission.Companion.DENIED: NotificationPermission get() =
"denied".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val
NotificationPermission.Companion.GRANTED: NotificationPermission get() =
"granted".asDynamic().unsafeCast<NotificationPermission>()\n\n/* please, don't implement this interface!\n
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface NotificationDirection {\n companion object\n}\n\npublic inline val
NotificationDirection.Companion.AUTO: NotificationDirection get() =
"auto".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.LTR: NotificationDirection get()
= "ltr".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.RTL: NotificationDirection get() =
"rtl".asDynamic().unsafeCast<NotificationDirection>()"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be

```

found in the license/LICENSE.txt file.\n \*\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n//  
See github.com/kotlin/dukat for details\n\npackage org.w3c.workers\n\nimport kotlin.js.\*\nimport  
org.khronos.webgl.\*\nimport org.w3c.dom.\*\nimport org.w3c.dom.events.\*\nimport org.w3c.fetch.\*\nimport  
org.w3c.notifications.\*\n\n/\*\*\n \* Exposes the JavaScript  
[ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin\n \*\n\npublic external  
abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker,  
UnionClientOrMessagePortOrServiceWorker {\n open val scriptURL:  
String\n open val state: ServiceWorkerState\n open var onstatechange: ((Event) -> dynamic)?\n fun  
postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/\*\*\n \* Exposes the JavaScript  
[ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to  
Kotlin\n \*\n\npublic external abstract class ServiceWorkerRegistration : EventTarget {\n open val installing:  
ServiceWorker?\n open val waiting: ServiceWorker?\n open val active: ServiceWorker?\n open val scope:  
String\n open var onupdatefound: ((Event) -> dynamic)?\n open val APISpace: dynamic\n fun update():  
Promise<Unit>\n fun unregister(): Promise<Boolean>\n fun showNotification(title: String, options:  
NotificationOptions = definedExternally): Promise<Unit>\n fun getNotifications(filter: GetNotificationOptions =  
definedExternally): Promise<Array<Notification>>\n fun methodName(): Promise<dynamic>\n}\n\n/\*\*\n \*  
Exposes the JavaScript  
[ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin\n\n  
\*\n\npublic external abstract class ServiceWorkerContainer : EventTarget {\n open val controller:  
ServiceWorker?\n open val ready: Promise<ServiceWorkerRegistration>\n open var oncontrollerchange:  
((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun register(scriptURL: String,  
options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>\n fun  
getRegistration(clientURL: String = definedExternally): Promise<Any?>\n fun getRegistrations():  
Promise<Array<ServiceWorkerRegistration>>\n fun startMessages()\n}\n\n\npublic external interface  
RegistrationOptions {\n var scope: String?\n get() = definedExternally\n set(value) = definedExternally\n  
var type: WorkerType? /\* = WorkerType.CLASSIC \*/\n get() = definedExternally\n set(value) =  
definedExternally\n}\n\n\n@Suppress(\"INVISIBLE\_REFERENCE\",  
\"INVISIBLE\_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun RegistrationOptions(scope: String? =  
undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {\n val o = js(\"{\}\")\n  
o[\"scope\"] = scope\n o[\"type\"] = type\n return o\n}\n\n\n/\*\*\n \* Exposes the JavaScript  
[ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to  
Kotlin\n\n \*\n\npublic external open class ServiceWorkerMessageEvent(type: String, eventInitDict:  
ServiceWorkerMessageEventInit = definedExternally) : Event {\n open val data: Any?\n open val origin:  
String\n open val lastEventId: String\n open val source: UnionMessagePortOrServiceWorker?\n open val  
ports: Array<out MessagePort>?\n\n companion object {\n val NONE: Short\n val  
CAPTURING\_PHASE: Short\n val AT\_TARGET: Short\n val BUBBLING\_PHASE: Short\n  
}\n}\n\n\npublic external interface ServiceWorkerMessageEventInit : EventInit {\n  
var data: Any?\n get() = definedExternally\n set(value) = definedExternally\n var origin: String?\n  
get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String?\n get() =  
definedExternally\n set(value) = definedExternally\n var source: UnionMessagePortOrServiceWorker?\n  
get() = definedExternally\n set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() =  
definedExternally\n set(value) = definedExternally\n}\n\n\n@Suppress(\"INVISIBLE\_REFERENCE\",  
\"INVISIBLE\_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun ServiceWorkerMessageEventInit(data:  
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:  
UnionMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? =  
false, cancelable: Boolean? = false, composed: Boolean? = false): ServiceWorkerMessageEventInit {\n val o =  
js(\"{\}\")\n}\n

```

 o["data"] = data\n o["origin"] = origin\n o["lastEventId"] = lastEventId\n o["source"] = source\n
 o["ports"] = ports\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] =
 composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to
Kotlin\n */\npublic external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {\n open val clients:
Clients\n open val registration: ServiceWorkerRegistration\n open var oninstall: ((Event) -> dynamic)?\n open
var onactivate: ((Event) -> dynamic)?\n open var onfetch: ((FetchEvent) -> dynamic)?\n open var
onforeignfetch: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n open var
onnotificationclick: ((NotificationEvent) -> dynamic)?\n open var onnotificationclose: ((NotificationEvent) ->
dynamic)?\n open var onfunctionalevent: ((Event)
-> dynamic)?\n fun skipWaiting(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin\n */\npublic external abstract class Client :
UnionClientOrMessagePortOrServiceWorker {\n open val url: String\n open val frameType: FrameType\n
open val id: String\n fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n
* Exposes the JavaScript [WindowClient](https://developer.mozilla.org/en/docs/Web/API/WindowClient) to
Kotlin\n */\npublic external abstract class WindowClient : Client {\n open val visibilityState: dynamic\n open
val focused: Boolean\n fun focus(): Promise<WindowClient>\n fun navigate(url: String):
Promise<WindowClient>\n}\n\n/**\n * Exposes the JavaScript
[Clients](https://developer.mozilla.org/en/docs/Web/API/Clients) to Kotlin\n */\npublic external abstract class
Clients {\n fun get(id: String): Promise<Any?>\n fun matchAll(options: ClientQueryOptions
= definedExternally): Promise<Array<Client>>\n fun openWindow(url: String): Promise<WindowClient?>\n
fun claim(): Promise<Unit>\n}\n\npublic external interface ClientQueryOptions {\n var includeUncontrolled:
Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var type:
ClientType? /* = ClientType.WINDOW */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW):
ClientQueryOptions {\n val o = js(\"({})\")\n o[\"includeUncontrolled\"] = includeUncontrolled\n o[\"type\"] =
type\n return o\n}\n\n/**\n * Exposes the JavaScript
[ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin\n */\npublic external
open class ExtendableEvent(type: String, eventInitDict:
ExtendableEventInit = definedExternally) : Event {\n fun waitUntil(f: Promise<Any?>)\n\n companion object
{\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n\n public external interface ExtendableEventInit :
EventInit\n\n @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n @kotlin.internal.InlineOnly\n public inline fun ExtendableEventInit(bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableEventInit {\n val o =
js(\"({})\")\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n
return o\n}\n\n/**\n * Exposes the JavaScript
[InstallEvent](https://developer.mozilla.org/en/docs/Web/API/InstallEvent) to Kotlin\n */\npublic external open
class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {\n fun
registerForeignFetch(options: ForeignFetchOptions)\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET:
Short\n val BUBBLING_PHASE: Short\n }\n\n public external interface ForeignFetchOptions {\n var
scopes: Array<String>?\n var origins: Array<String>?\n }\n\n @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n @kotlin.internal.InlineOnly\n public inline fun ForeignFetchOptions(scopes:
Array<String>?, origins: Array<String>?): ForeignFetchOptions {\n val o = js(\"({})\")\n o[\"scopes\"] =
scopes\n o[\"origins\"] = origins\n return o\n}\n\n/**\n * Exposes the JavaScript

```

```

[FetchEvent](https://developer.mozilla.org/en/docs/Web/API/FetchEvent) to Kotlin\n *\npublic external open class
FetchEvent(type: String, eventInitDict: FetchEventInit) : ExtendableEvent {\n open val request: Request\n open
val clientId: String?\n open val isReload: Boolean\n fun respondWith(r: Promise<Response>)\n\n companion
object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface FetchEventInit : ExtendableEventInit {\n var request: Request?\n var
clientId: String? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n var isReload:
Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun FetchEventInit(request: Request?,
clientId: String? = null, isReload: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): FetchEventInit {\n val o = js(\"({})\")\n o[\"request\"] = request\n o[\"clientId\"]
= clientId\n o[\"isReload\"] = isReload\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n
o[\"composed\"] = composed\n return o\n}\n\npublic external
open class ForeignFetchEvent(type: String, eventInitDict: ForeignFetchEventInit) : ExtendableEvent {\n open val
request: Request\n open val origin: String\n fun respondWith(r: Promise<ForeignFetchResponse>)\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET:
Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ForeignFetchEventInit :
ExtendableEventInit {\n var request: Request?\n var origin: String? /* = \"null\" */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchEventInit(request:
Request?, origin: String? = \"null\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): ForeignFetchEventInit {\n val o = js(\"({})\")\n o[\"request\"] = request\n o[\"origin\"] = origin\n
o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\npublic external interface
ForeignFetchResponse {\n var response: Response?\n var origin: String?\n get() = definedExternally\n
set(value) = definedExternally\n var headers: Array<String>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchResponse(response:
Response?, origin: String? = undefined, headers: Array<String>? = undefined): ForeignFetchResponse {\n val o =
js(\"({})\")\n o[\"response\"] = response\n o[\"origin\"] = origin\n o[\"headers\"] = headers\n return
o\n}\n\n/**\n * Exposes the JavaScript
[ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin\n
*\npublic external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit
= definedExternally) : ExtendableEvent {\n open val data: Any?\n open val origin: String\n open val
lastEventId: String\n open val source: UnionClientOrMessagePortOrServiceWorker?\n open val ports:
Array<out MessagePort>?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
ExtendableMessageEventInit : ExtendableEventInit {\n var data: Any?\n get() = definedExternally\n
set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n set(value) =
definedExternally\n var lastEventId: String?\n get() = definedExternally\n set(value) =
definedExternally\n var source: UnionClientOrMessagePortOrServiceWorker?\n get() = definedExternally\n
set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() = definedExternally\n
set(value)
= definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionClientOrMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles:

```

```

Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableMessageEventInit {\n
val o = js("{}")\n o["data"] = data\n o["origin"] = origin\n o["lastEventId"] = lastEventId\n
o["source"] = source\n o["ports"] = ports\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n
o["composed"] = composed\n return o}\n\n/**\n * Exposes the JavaScript
[Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin\n */\npublic external abstract class Cache
{\n fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>\n
fun matchAll(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Response>>\n fun add(request: dynamic): Promise<Unit>\n fun addAll(requests:
Array<dynamic>): Promise<Unit>\n fun put(request: dynamic, response: Response): Promise<Unit>\n fun
delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>\n fun
keys(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Request>>\n}\n\npublic external interface CacheQueryOptions {\n var ignoreSearch: Boolean? /*
= false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreMethod: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreVary: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var cacheName: String?\n
get() = definedExternally\n
set(value) = definedExternally}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CacheQueryOptions(ignoreSearch:
Boolean? = false, ignoreMethod: Boolean? = false, ignoreVary: Boolean? = false, cacheName: String? = undefined):
CacheQueryOptions {\n val o = js("{}")\n o["ignoreSearch"] = ignoreSearch\n o["ignoreMethod"] =
ignoreMethod\n o["ignoreVary"] = ignoreVary\n o["cacheName"] = cacheName\n return o}\n\npublic
external interface CacheBatchOperation {\n var type: String?\n get() = definedExternally\n set(value) =
definedExternally\n var request: Request?\n get() = definedExternally\n set(value) = definedExternally\n
var response: Response?\n get() = definedExternally\n set(value) = definedExternally\n var options:
CacheQueryOptions?\n get() = definedExternally\n set(value) =
definedExternally}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CacheBatchOperation(type: String? =
undefined, request: Request? = undefined, response: Response? = undefined, options: CacheQueryOptions? =
undefined): CacheBatchOperation {\n val o = js("{}")\n o["type"] = type\n o["request"] = request\n
o["response"] = response\n o["options"] = options\n return o}\n\n/**\n * Exposes the JavaScript
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage) to Kotlin\n */\npublic external
abstract class CacheStorage {\n fun match(request: dynamic, options: CacheQueryOptions = definedExternally):
Promise<Any?>\n fun has(cacheName: String): Promise<Boolean>\n fun open(cacheName: String):
Promise<Cache>\n fun delete(cacheName: String): Promise<Boolean>\n fun keys():
Promise<Array<String>>\n}\n\npublic external open class FunctionalEvent : ExtendableEvent {\n companion
object {\n val NONE: Short\n
val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface UnionMessagePortOrServiceWorker\n\npublic external interface
UnionClientOrMessagePortOrServiceWorker\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ServiceWorkerState {\n companion object\n}\n\npublic inline val
ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() =
"installing".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() =
"installed".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() =
"activating".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() =

```



```

\"activated\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic
inline val ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() =
\"redundant\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface FrameType {\n companion object\n}\n\npublic inline val FrameType.Companion.AUXILIARY:
FrameType get() = \"auxiliary\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val
FrameType.Companion.TOP_LEVEL: FrameType get() = \"top-
level\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NESTED: FrameType
get() = \"nested\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NONE:
FrameType get() = \"none\".asDynamic().unsafeCast<FrameType>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ClientType {\n companion object\n}\n\npublic
inline val ClientType.Companion.WINDOW: ClientType get() =
\"window\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.WORKER:
ClientType get() = \"worker\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val
ClientType.Companion.SHAREDWORKER: ClientType get() =
\"sharedworker\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.ALL:
ClientType get() = \"all\".asDynamic().unsafeCast<ClientType>()\", /*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.xhr\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.files.*\n\n/*\n *
Exposes the JavaScript
[XMLHttpRequestEventTarget](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestEventTarget)
to Kotlin\n\n *\n\npublic external abstract class XMLHttpRequestEventTarget : EventTarget {\n open var
onloadstart: ((ProgressEvent) -> dynamic)?\n open var onprogress: ((ProgressEvent) -> dynamic)?\n open var
onabort: ((Event) -> dynamic)?\n open var onerror: ((Event) -> dynamic)?\n open var onload: ((Event) ->
dynamic)?\n open var ontimeout: ((Event) -> dynamic)?\n open var onloadend: ((Event) ->
dynamic)?\n}\n\n\npublic external abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget\n\n/**\n *
Exposes the JavaScript [XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to
Kotlin\n\n *\n\npublic external open class XMLHttpRequest : XMLHttpRequestEventTarget {\n var
onreadystatechange: ((Event) -> dynamic)?\n open val readyState: Short\n var timeout: Int\n var
withCredentials: Boolean\n open val upload: XMLHttpRequestUpload\n open val responseURL: String\n
open val status: Short\n open val statusText: String\n var responseType: XMLHttpRequestResponseType\n
open val response: Any?\n open val responseText: String\n open val responseXML: Document?\n fun
open(method: String, url: String)\n fun open(method: String, url: String, async: Boolean, username: String? =
definedExternally, password: String? = definedExternally)\n fun setRequestHeader(name: String, value: String)\n
fun send(body: dynamic = definedExternally)\n fun abort()\n fun getResponseHeader(name: String): String?\n
fun getAllResponseHeaders(): String\n fun overrideMimeType(mime: String)\n\n companion object {\n val
UNSENT: Short\n val OPENED: Short\n val HEADERS_RECEIVED: Short\n val LOADING:
Short\n val DONE: Short\n }\n}\n\n\n/**\n * Exposes the JavaScript
[FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin\n\n *\n\npublic external open class
FormData(form: HTMLFormElement
= definedExternally) {\n fun append(name: String, value: String)\n fun append(name: String, value: Blob,
filename: String = definedExternally)\n fun delete(name: String)\n fun get(name: String): dynamic\n fun
getAll(name: String): Array<dynamic>\n fun has(name: String): Boolean\n fun set(name: String, value: String)\n
fun set(name: String, value: Blob, filename: String = definedExternally)\n}\n\n\n/**\n * Exposes the JavaScript
[ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin\n\n *\n\npublic external open

```

```

class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {
 open val lengthComputable: Boolean
 open val loaded: Number
 open val total: Number
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external interface ProgressEventInit : EventInit {
 var lengthComputable: Boolean? /* = false */
 get() = definedExternally
 set(value) = definedExternally
 var loaded: Number? /* = 0 */
 get() = definedExternally
 set(value) = definedExternally
 var total: Number? /* = 0 */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ProgressEventInit(lengthComputable: Boolean? = false, loaded: Number? = 0, total: Number? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ProgressEventInit {
 val o = js("{}")
 o["lengthComputable"] = lengthComputable
 o["loaded"] = loaded
 o["total"] = total
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

/* please, don't implement this interface! */
@JsName("null")
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface XMLHttpRequestResponseType {
 companion object {
 public inline val XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() = ""
 public inline val XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() = "arraybuffer"
 public inline val XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() = "blob"
 public inline val XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() = "document"
 public inline val XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() = "json"
 public inline val XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() = "text"
 }
}

/* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin
import kotlin.annotation.AnnotationRetention.BINARY
import kotlin.annotation.AnnotationRetention.SOURCE
import kotlin.annotation.AnnotationTarget.*
import kotlin.internal.RequireKotlin
import kotlin.internal.RequireKotlinVersionKind
import kotlin.reflect.KClass

/** Signals that the annotated annotation class is a marker of an experimental API.
 * Any declaration annotated with that marker is considered an experimental declaration
 * and its call sites should accept the experimental aspect of it either by using [UseExperimental],
 * or by being annotated with that marker themselves, effectively causing further propagation of that experimental aspect.
 * This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].
 */
@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.2")
@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6")
@Deprecated("Please use RequiresOptIn instead.")
public annotation class Experimental(val level: Level = Level.ERROR) {
 /** Severity of the diagnostic that should be reported on usages of experimental API which did not explicitly accept the experimental aspect
 * of that API either by using [UseExperimental] or by being annotated with the corresponding marker annotation.
 */
 public enum class Level {
 /** Specifies that a warning should be reported on incorrect usages of this experimental API.
 */
 WARNING,
 /** Specifies that an error should be reported on incorrect usages of this experimental API.
 */
 ERROR,
 }
}

```

```

}
}

/**
 * Allows to use experimental API denoted by the given markers in the annotated file, declaration, or
 * expression.
 * If a declaration is annotated with [UseExperimental], its usages are not required to opt-in to that
 * experimental API.
 * This class is deprecated in favor of a more general approach provided by
 * [RequiresOptIn]/[OptIn].
 * @Target(CLASS, PROPERTY, LOCAL_VARIABLE,
 * VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER,
 * EXPRESSION, FILE,
 * TYPEALIAS)
 * @Retention(SOURCE)
 * @SinceKotlin("1.2")
 * @RequireKotlin("1.2.50", versionKind =
 * RequireKotlinVersionKind.COMPILER_VERSION)
 * @DeprecatedSinceKotlin(warningSince = "1.4", errorSince
 * = "1.6")
 * @Deprecated("Please use OptIn instead.", ReplaceWith("OptIn(*markerClass)",
 * "kotlin.OptIn"))
 * public annotation class UseExperimental(
 * vararg val markerClass: KClass<out
 * Annotation>)
 * @Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
 * TYPEALIAS)
 * @Retention(BINARY)
 * internal
 * annotation class WasExperimental(
 * vararg val markerClass: KClass<out Annotation>)
 * "package
 * kotlin
 * import kotlin.annotation.AnnotationTarget.*
 * This annotation marks the standard library API
 * that is considered experimental and is not subject to the
 * [general compatibility
 * guarantees](https://kotlinlang.org/docs/reference/evolution/components-stability.html) given for the standard
 * library:
 * the behavior of such API may be changed or the API may be removed completely in any further
 * release.
 * > Beware using the annotated API especially if you're developing a library, since your library might
 * become binary incompatible
 * with the future versions of the standard library.
 * Any usage of a declaration
 * annotated with `@ExperimentalStdlibApi` must be accepted either by
 * annotating that usage with the [OptIn]
 * annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`,
 * or by using the compiler argument `--opt-
 * in=kotlin.ExperimentalStdlibApi`.
 * @RequiresOptIn(level =
 * RequiresOptIn.Level.ERROR)
 * @Retention(AnnotationRetention.BINARY)
 * @Target(CLASS,
 * ANNOTATION_CLASS,
 * PROPERTY,
 * FIELD,
 * LOCAL_VARIABLE,
 * VALUE_PARAMETER,
 * CONSTRUCTOR,
 * FUNCTION,
 * PROPERTY_GETTER,
 * PROPERTY_SETTER,
 * TYPEALIAS)
 * @MustBeDocumented
 * @SinceKotlin("1.3")
 * public annotation class
 * ExperimentalStdlibApi
 * "/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 * @BuilderInference
 * @BuilderInference
 * @BuilderInference
 * Allows to infer generic type arguments of a function
 * from the calls in the annotated function parameter of that function.
 * When this annotation is placed on a
 * generic function parameter of a function,
 * it enables to infer the type arguments of
 * that generic function from the lambda body passed to that parameter.
 * The calls that affect inference are
 * either members of the receiver type of an annotated function parameter or
 * extensions for that type. The
 * extensions must be themselves annotated with `@BuilderInference`.
 * Example: we declare
 * fun
 * <T> sequence(@BuilderInference block: suspend SequenceScope<T>().->Unit): Sequence<T>
 * and use
 * it like
 * val result = sequence { yield("result") }
 * Here the type argument of the resulting
 * sequence is inferred to `String` from
 * the argument of the [SequenceScope.yield] function, that is called inside
 * the lambda passed to [sequence].
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on
 * how to opt-in for it.
 * @Target(VALUE_PARAMETER, FUNCTION,
 * PROPERTY)
 * @Retention(AnnotationRetention.BINARY)
 * @SinceKotlin("1.3")
 * @ExperimentalTypeInferenc
 * e
 * public annotation class BuilderInference
 *
 * Enables overload selection based on the type of the value returned from lambda argument.
 * When two or
 * more function overloads have otherwise the same parameter lists that differ only in the return type
 * of a
 * functional parameter, this annotation enables overload selection by the type of the value returned from
 * the
 * lambda function passed to this functional parameter.
 * Example:
 * @OverloadResolutionByLambdaReturnType
 * fun create(intProducer: () -> Int): Int
 * fun

```

create(doubleProducer: () -> Double): Double\n \* val newValue = create { 3.14 }\n \* The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned from the lambda function.\n \* This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported.\n

\* Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.\n

```
/\n@Target(FUNCTION)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.4")\n@ExperimentalTypeInference\npublic annotation class OverloadResolutionByLambdaReturnType", /\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget\n\nimport kotlin.internal.RequireKotlin\n\nimport kotlin.internal.RequireKotlinVersionKind\n\n/**\n * The experimental multiplatform support API marker.\n *\n * Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalMultiplatform::class)`\n * or by using the compiler argument `-opt-in=kotlin.ExperimentalMultiplatform`.\n
```

```
*/\n@RequiresOptIn\n@MustBeDocumented\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n TYPEALIAS\n)\n@Retention(AnnotationRetention.BINARY)\n@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class ExperimentalMultiplatform\n\n/**\n * Marks an expected annotation class that it isn't required to have actual counterparts in all platforms.\n *\n * This annotation is only applicable to `expect` annotation classes in multi-platform projects and marks that class as\n * `optional`.\n *\n * Optional expected class is allowed to have no corresponding actual class on the platform. Optional annotations can only be used\n * to annotate something, not as types in signatures. If an optional annotation has no corresponding actual class on a platform,\n * the annotation entries where it's used are simply erased when compiling code on that platform.\n *\n * Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.\n
```

```
/\n@Target(ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@ExperimentalMultiplatform\n@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class OptionalExpectation\n", /\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin\n\nimport kotlin.annotation.AnnotationRetention.BINARY\n\nimport kotlin.annotation.AnnotationRetention.SOURCE\n\nimport kotlin.annotation.AnnotationTarget\n\nimport kotlin.internal.RequireKotlin\n\nimport kotlin.internal.RequireKotlinVersionKind\n\nimport kotlin.reflect.KClass\n\n/**\n * Signals that the annotated annotation class is a marker of an API that requires an explicit opt-in.\n *\n * Call sites of any declaration annotated with that marker should opt in to the API either by using [OptIn],\n * or by being annotated with that marker themselves, effectively causing further propagation of the opt-in requirement.\n *\n * This class requires opt-in itself and can only be used with the compiler argument\n * -opt-in=kotlin.RequiresOptIn.\n *\n * @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.\n * The default message is: `\"This declaration is experimental and its usage should be marked with 'Marker'` or `@OptIn(Marker::class)`,\n * where `Marker` is the opt-in requirement marker.\n * @property level specifies how usages of API without an explicit opt-in are reported in code.\n
```

```
*/\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.3")\n@RequireKotlin("1.3.70", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class RequiresOptIn(\n val message: String = "",\n val level: Level = Level.ERROR\n) {\n /**\n * Severity of the diagnostic that should be reported on usages which did not explicitly opt into\n * the API either by using [OptIn] or by being
```

```

annotated with the corresponding marker annotation.\n *\n public enum class Level {\n /** Specifies that a
warning should be reported on incorrect usages of this API. */\n WARNING,\n /** Specifies that an error
should be reported on incorrect usages of this API. */\n ERROR,\n }\n}\n\n/**\n * Allows to use the API
denoted by the given markers in the annotated file, declaration, or expression.\n * If a declaration is annotated with
[OptIn], its usages are **not** required to opt in to that API.\n *\n * This class requires opt-in itself and can only be
used with the compiler argument -opt-in=kotlin.RequiresOptIn.\n *\n @Target(\n CLASS, PROPERTY,
LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION,
PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE,
TYPEALIAS)\n @Retention(SOURCE)\n @SinceKotlin("1.3")\n @RequireKotlin("1.3.70", versionKind =
RequireKotlinVersionKind.COMPILER_VERSION)\n public annotation class OptIn(\n vararg val markerClass:
KClass<out Annotation>)\n }\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n package kotlin.collections\n import kotlin.js.JsName\n\n /**\n * Provides a skeletal
implementation of the read-only [Collection] interface.\n *\n * @param E the type of elements contained in the
collection. The collection is covariant in its element type.\n *\n @SinceKotlin("1.1")\n public abstract class
AbstractCollection<out E> protected constructor() : Collection<E> {\n abstract override val size: Int\n abstract
override fun iterator(): Iterator<E>\n\n override fun contains(element:
@UnsafeVariance E): Boolean = any { it == element }\n\n override fun containsAll(elements:
Collection<@UnsafeVariance E>): Boolean =\n elements.all { contains(it) } // use when js will support bound
refs: elements.all(this::contains)\n\n override fun isEmpty(): Boolean = size == 0\n\n override fun toString():
String = joinToString(", ", "[", "]") {\n if (it === this) "(this Collection)" else it.toString()\n }\n}\n\n /**\n * Returns new array of type Array<Any?> with the elements of this collection.\n *\n @JsName("toArray")\n protected open fun toArray(): Array<Any?> = copyToArrayImpl(this)\n\n /**\n *
Fills the provided [array] or creates new array of the same type\n * and fills it with the elements of this
collection.\n *\n protected open fun <T> toArray(array: Array<T>): Array<T> = copyToArrayImpl(this,
array)\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n package kotlin.collections\n\n private enum class State {\n Ready,\n NotReady,\n Done,\n Failed\n}\n\n /**\n * A base class to simplify implementing iterators so that implementations only have to implement
[computeNext]\n * to implement the iterator, calling [done] when the iteration is complete.\n *\n public abstract
class AbstractIterator<T> : Iterator<T> {\n private var state = State.NotReady\n private var nextValue: T? =
null\n\n override fun hasNext(): Boolean {\n require(state != State.Failed)\n return when (state) {\n
State.Done -> false\n State.Ready -> true\n else -> tryToComputeNext()\n }\n }\n\n override
fun next(): T {\n if (!hasNext()) throw NoSuchElementException()\n state = State.NotReady\n
@Suppress("UNCHECKED_CAST")\n return nextValue as T\n
}\n\n private fun tryToComputeNext(): Boolean {\n state = State.Failed\n computeNext()\n return
state == State.Ready\n }\n\n /**\n * Computes the next item in the iterator.\n *\n * This callback method
should call one of these two methods:\n * * [setNext] with the next value of the iteration\n * * [done] to
indicate there are no more elements\n * * Failure to call either method will result in the iteration terminating
with a failed state\n *\n abstract protected fun computeNext(): Unit\n\n /**\n * Sets the next value in the
iteration, called from the [computeNext] function\n *\n protected fun setNext(value: T): Unit {\n nextValue = value\n state = State.Ready\n }\n\n /**\n * Sets the state to done so that the iteration
terminates.\n *\n protected fun done() {\n state = State.Done\n }\n}\n\n", /*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n package kotlin.collections\n\n /**\n * Based on GWT AbstractList\n * Copyright 2007 Google
Inc.\n *\n package kotlin.collections\n\n /**\n * Provides a skeletal implementation of the read-only [List]
interface.\n *\n * This class is intended to help implementing read-only lists so it doesn't support concurrent

```

```

modification tracking.\n * \n * @param E the type of elements contained in the list. The list is covariant in its
element type.\n * \n @SinceKotlin("1.1")\npublic abstract class AbstractList<out E> protected constructor() :
AbstractCollection<E>(), List<E> {\n abstract override val size: Int\n abstract override fun get(index: Int): E\n\n override fun iterator(): Iterator<E> = IteratorImpl()\n\n override fun indexOf(element: @UnsafeVariance E): Int
= indexOfFirst { it == element }\n\n override fun lastIndexOf(element: @UnsafeVariance E): Int
= indexOfLast { it == element }\n\n override fun listIterator(): ListIterator<E> = ListIteratorImpl(0)\n\n override fun listIterator(index: Int): ListIterator<E> = ListIteratorImpl(index)\n\n override fun subList(fromIndex:
Int, toIndex: Int): List<E> = SubList(this, fromIndex, toIndex)\n\n private class SubList<out E>(private val list:
AbstractList<E>, private val fromIndex: Int, toIndex: Int) : AbstractList<E>(), RandomAccess {\n private var
_size: Int = 0\n\n init {\n checkRangeIndexes(fromIndex, toIndex, list.size)\n this._size = toIndex
- fromIndex\n }\n\n override fun get(index: Int): E {\n checkElementIndex(index, _size)\n\n return list[fromIndex + index]\n }\n\n override val size: Int get() = _size\n }\n\n /**\n * Compares
this list with other list instance with the ordered structural equality.\n * \n * @return true, if [other] instance is a
[List] of the same size,
which contains the same elements in the same order.\n * \n * override fun equals(other: Any?): Boolean {\n
if (other === this) return true\n if (other !is List<*>) return false\n return orderedEquals(this, other)\n
}\n\n /**\n * Returns the hash code value for this list.\n * \n * override fun hashCode(): Int =
orderedHashCode(this)\n\n private open inner class IteratorImpl : Iterator<E> {\n /** the index of the item
that will be returned on the next call to [next]`()\n * \n * protected var index = 0\n * \n * override fun hasNext():
Boolean = index < size\n * \n * override fun next(): E {\n * \n * if (!hasNext()) throw NoSuchElementException()\n
 * \n * return get(index++)\n * \n * }\n }\n\n /**\n * Implementation of [ListIterator] for abstract lists.\n * \n * \n * private open inner class ListIteratorImpl(index: Int) : IteratorImpl(), ListIterator<E> {\n * \n * init {\n
checkPositionIndex(index,
this@AbstractList.size)\n * \n * this.index = index\n * \n * override fun hasNext(): Boolean = index >
0\n * \n * override fun nextIndex(): Int = index\n * \n * override fun previous(): E {\n * \n * if (!hasPrevious())
throw NoSuchElementException()\n * \n * return get(--index)\n * \n * }\n * \n * override fun previousIndex(): Int =
index - 1\n * \n * }\n\n internal companion object {\n internal fun checkElementIndex(index: Int, size: Int) {\n
if (index < 0 || index >= size) {\n throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n }\n }\n\n internal fun checkPositionIndex(index: Int, size: Int) {\n if (index < 0 || index > size) {\n
throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n }\n }\n\n internal fun
checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {\n if (fromIndex < 0 || toIndex > size)\n {\n throw IndexOutOfBoundsException("\nfromIndex: $fromIndex, toIndex: $toIndex, size: $size")\n }\n if (fromIndex > toIndex) {\n throw IllegalArgumentException("\nfromIndex: $fromIndex >
toIndex: $toIndex")\n }\n }\n\n internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size:
Int) {\n if (startIndex < 0 || endIndex > size) {\n throw IndexOutOfBoundsException("\nstartIndex:
$startIndex, endIndex: $endIndex, size: $size")\n }\n if (startIndex > endIndex) {\n throw
IllegalArgumentException("\nstartIndex: $startIndex > endIndex: $endIndex")\n }\n }\n\n internal
fun orderedHashCode(c: Collection<*>): Int {\n var hashCode = 1\n for (e in c) {\n hashCode = 31 * hashCode + (e?.hashCode() ?: 0)\n }\n return hashCode\n }\n\n internal fun
orderedEquals(c:
Collection<*>, other: Collection<*>): Boolean {\n if (c.size != other.size) return false\n\n val
otherIterator = other.iterator()\n for (elem in c) {\n val elemOther = otherIterator.next()\n if
(elem != elemOther) {\n return false\n }\n }\n return true\n }\n\n }"/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n * \n * \n * Based on
GWT AbstractMap\n * Copyright 2007 Google Inc.\n * \n * \n * package kotlin.collections\n * \n * /**\n * Provides a skeletal
implementation of the read-only [Map] interface.\n * \n * \n * * The implementor is required to implement [entries]
property, which should return read-only set of map entries.\n * \n * \n * @param K the type of map keys. The map is

```

invariant in its key type.\n \* @param

```
V the type of map values. The map is covariant in its value type.\n *^/\n@SinceKotlin("1.1")\npublic abstract class\nAbstractMap<K, out V> protected constructor() : Map<K, V> {\n\n override fun containsKey(key: K): Boolean\n {\n return implFindEntry(key) != null\n }\n\n override fun containsValue(value: @UnsafeVariance V):\n Boolean = entries.any { it.value == value }\n\n internal fun containsEntry(entry: Map.Entry<*, *>): Boolean {\n // since entry comes from @UnsafeVariance parameters it can be virtually anything\n if (entry !is\n Map.Entry<*, *>) return false\n val key = entry.key\n val value = entry.value\n val ourValue =\n get(key)\n if (value != ourValue) {\n return false\n }\n // Perhaps it was null and we don't\n contain the key?\n if (ourValue == null && !containsKey(key)) {\n return false\n }\n return\n true\n }\n\n /**\n * Compares this map with other\n instance with the ordered structural equality.\n *\n * @return true, if [other] instance is a [Map] of the same\n size, all entries of which are contained in the [entries] set of this map.\n */\n override fun equals(other: Any?):\n Boolean {\n if (other === this) return true\n if (other !is Map<*, *>) return false\n if (size != other.size)\n return false\n return other.entries.all { containsEntry(it) }\n }\n\n override operator fun get(key: K): V? =\n implFindEntry(key)?.value\n\n /**\n * Returns the hash code value for this map.\n *\n * It is the same as\n the hashCode of [entries] set.\n */\n override fun hashCode(): Int = entries.hashCode()\n\n override fun\n isEmpty(): Boolean = size == 0\n\n override val size: Int get() = entries.size\n\n /**\n * Returns a read-only\n [Set] of all keys in this map.\n *\n * Accessing this property first time creates a keys view from [entries].\n *\n * All subsequent accesses\n just return the created instance.\n */\n override val keys: Set<K>\n get() {\n if (_keys == null) {\n _keys = object : AbstractSet<K>() {\n override operator fun contains(element: K): Boolean =\n containsKey(element)\n override operator fun iterator(): Iterator<K> {\n val\n entryIterator = entries.iterator()\n return object : Iterator<K> {\n override fun\n hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): K = entryIterator.next().key\n }\n }\n override val size: Int get() = this@AbstractMap.size\n }\n return _keys!!\n }\n @kotlin.jvm.Volatile\n private var _keys: Set<K>? = null\n }\n\n override fun toString(): String = entries.joinToString(", ", "{", "}") { toString(it)\n }\n\n private fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" +\n toString(entry.value)\n\n private fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()\n\n /**\n * Returns a read-only [Collection] of all values in this map.\n *\n * Accessing this property first time\n creates a values view from [entries].\n *\n * All subsequent accesses just return the created instance.\n */\n override val values: Collection<V>\n get() {\n if (_values == null) {\n _values = object :\n AbstractCollection<V>() {\n override operator fun contains(element: @UnsafeVariance V): Boolean =\n containsValue(element)\n override operator fun iterator(): Iterator<V> {\n val\n entryIterator = entries.iterator()\n return object : Iterator<V> {\n override fun\n hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): V = entryIterator.next().value\n }\n }\n }\n return _values!!\n }\n @kotlin.jvm.Volatile\n private var _values: Collection<V>? = null\n private fun implFindEntry(key: K):\n Map.Entry<K, V>? = entries.firstOrNull { it.key == key }\n\n internal companion object {\n internal fun\n entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?: 0) xor (value?.hashCode() ?: 0) }\n internal fun\n entryToString(e: Map.Entry<*, *>): String = with(e) { "$key=$value" }\n internal fun\n entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {\n if (other !is Map.Entry<*, *>) return false\n return e.key == other.key && e.value == other.value\n }\n }\n }\n\n /**\n * Copyright 2010-2020 JetBrains\n s.r.o. and Kotlin
```

Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be\n found in the license/LICENSE.txt file.\n \*/\npackage kotlin.collections\n\n/\*\*\n \* Provides a skeletal\n implementation of the read-only [Set] interface.\n \*\n \* This class is intended to help implementing read-only sets so

it doesn't support concurrent modification tracking.\n \* \n \* @param E the type of elements contained in the set. The set is covariant in its element type.\n \* \n @SinceKotlin("1.1")\n public abstract class AbstractSet<out E> protected constructor() : AbstractCollection<E>(), Set<E> {\n \n /\*\*\n \* Compares this set with other set instance with the unordered structural equality.\n \* \n \* @return true, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.\n \* \n \* override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is Set<\*>) return false\n return setEquals(this, other)\n }\n \n /\*\*\n \* Returns the hash code value for this set.\n \* \n \* override fun hashCode(): Int = unorderedHashCode(this)\n \n internal companion object {\n internal fun unorderedHashCode(c: Collection<\*>): Int {\n var hashCode = 0\n for (element in c) {\n hashCode += (element?.hashCode() ?: 0)\n }\n return hashCode\n }\n \n internal fun setEquals(c: Set<\*>, other: Set<\*>): Boolean {\n if (c.size != other.size) return false\n return c.containsAll(other)\n }\n }\n \n }", "/\*\*\n \* Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \* \n package kotlin.collections\n /\*\*\n \* Resizable-array implementation of the deque data structure.\n \* \n \* The name deque is short for "double ended queue" and is usually pronounced "deck".\n \* \n \* The collection provide methods for convenient access to the both ends.\n \* \n \* It also implements [MutableList] interface and supports efficient get/set operations by index.\n \* \n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public class ArrayDeque<E> : AbstractMutableList<E> {\n private var head: Int = 0\n private var elementData: Array<Any?>\n \n override var size: Int = 0\n private set\n /\*\*\n \* Constructs an empty deque with specified [initialCapacity], or throws [IllegalArgumentException] if [initialCapacity] is negative.\n \* \n \* public constructor(initialCapacity: Int) {\n elementData = when {\n initialCapacity == 0 -> emptyElementData\n initialCapacity > 0 -> arrayOfNulls(initialCapacity)\n else -> throw IllegalArgumentException("Illegal Capacity: \$initialCapacity")\n }\n }\n \n /\*\*\n \* Constructs an empty deque.\n \* \n \* public constructor() {\n elementData = emptyElementData\n }\n \n /\*\*\n \* Constructs a deque that contains the same elements as the specified [elements] collection in the same order.\n \* \n \* public constructor(elements: Collection<E>) {\n elementData = elements.toArray()\n size = elementData.size\n if (elementData.isEmpty()) elementData = emptyElementData\n }\n \n /\*\*\n \* Ensures that the capacity of this deque is at least equal to the specified [minCapacity].\n \* \n \* If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.\n \* \n \* Otherwise, this method takes no action and simply returns.\n \* \n \* private fun ensureCapacity(minCapacity: Int) {\n if (minCapacity < 0) throw IllegalStateException("Deque is too big.") // overflow\n if (minCapacity <= elementData.size) return\n if (elementData === emptyElementData) {\n elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))\n return\n }\n val newCapacity = newCapacity(elementData.size, minCapacity)\n copyElements(newCapacity)\n }\n \n /\*\*\n \* Creates a new array with the specified [newCapacity] size and copies elements in the [elementData] array to it.\n \* \n \* private fun copyElements(newCapacity: Int) {\n val newElements = arrayOfNulls<Any?>(newCapacity)\n elementData.copyInto(newElements, 0, head, elementData.size)\n elementData.copyInto(newElements, elementData.size - head, 0, head)\n head = 0\n elementData = newElements\n }\n \n @kotlin.internal.InlineOnly\n private inline fun internalGet(internalIndex: Int): E {\n @Suppress("UNCHECKED\_CAST")\n return elementData[internalIndex] as E\n }\n \n private fun positiveMod(index: Int): Int = if (index >= elementData.size) index - elementData.size else index\n \n private fun negativeMod(index: Int): Int = if (index < 0) index + elementData.size else index\n \n @kotlin.internal.InlineOnly\n private inline fun internalIndex(index: Int): Int = positiveMod(head + index)\n \n private fun incremented(index: Int): Int = if (index == elementData.lastIndex) 0 else index + 1\n \n private fun decremented(index: Int): Int = if (index == 0) elementData.lastIndex else index - 1\n \n override fun isEmpty(): Boolean = size == 0\n /\*\*\n \* Returns the first element, or throws [NoSuchElementException] if this deque is empty.\n \* \n \* public fun first(): E = if



```

(isEmpty()) throw NoSuchElementException("ArrayDeque is empty.") else internalGet(head)\n\n /**\n *
Returns the first element, or `null` if this deque is empty.\n\n *\n public fun firstOrNull(): E? = if (isEmpty()) null
else internalGet(head)\n\n /**\n * Returns the last element, or throws [NoSuchElementException] if this deque
is empty.\n\n *\n public fun last(): E = if
(isEmpty()) throw NoSuchElementException("ArrayDeque is empty.") else
internalGet(internalIndex(lastIndex))\n\n /**\n * Returns the last element, or `null` if this deque is empty.\n
*\n public fun lastOrNull(): E? = if (isEmpty()) null else internalGet(internalIndex(lastIndex))\n\n /**\n *
Prepends the specified [element] to this deque.\n\n *\n public fun addFirst(element: E) {\n
ensureCapacity(size + 1)\n\n head = decremented(head)\n\n elementData[head] = element\n\n size += 1\n
}\n\n /**\n * Appends the specified [element] to this deque.\n\n *\n public fun addLast(element: E) {\n
ensureCapacity(size + 1)\n\n elementData[internalIndex(size)] = element\n\n size += 1\n\n }\n\n /**\n *
Removes the first element from this deque and returns that removed element, or throws [NoSuchElementException]
if this deque is empty.\n\n *\n public fun removeFirst(): E {\n\n if (isEmpty()) throw
NoSuchElementException("ArrayDeque
is empty.")\n\n val element = internalGet(head)\n\n elementData[head] = null\n\n head =
incremented(head)\n\n size -= 1\n\n return element\n\n }\n\n /**\n * Removes the first element from this
deque and returns that removed element, or returns `null` if this deque is empty.\n\n *\n public fun
removeFirstOrNull(): E? = if (isEmpty()) null else removeFirst()\n\n /**\n * Removes the last element from this
deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty.\n\n *\n
public fun removeLast(): E {\n\n if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")\n\n
val internalLastIndex = internalIndex(lastIndex)\n\n val element = internalGet(internalLastIndex)\n
elementData[internalLastIndex] = null\n\n size -= 1\n\n return element\n\n }\n\n /**\n * Removes the last
element from this deque and returns that removed
element, or returns `null` if this deque is empty.\n\n *\n public fun removeLastOrNull(): E? = if (isEmpty()) null
else removeLast()\n\n // MutableList, MutableCollection\n\n public override fun add(element: E): Boolean {\n
addLast(element)\n\n return true\n\n }\n\n public override fun add(index: Int, element: E) {\n
AbstractList.checkPositionIndex(index, size)\n\n if (index == size) {\n\n addLast(element)\n
return\n\n } else if (index == 0) {\n\n addFirst(element)\n\n return\n\n }\n\n ensureCapacity(size
+ 1)\n\n // Elements in circular array lay in 2 ways:\n\n // 1. `head` is less than `tail`: [#, #, e1, e2, e3,
#]\n\n // 2. `head` is greater than `tail`: [e3, #, #, #, e1, e2]\n\n // where head is the index of the first element
in the circular array,\n\n // and tail is the index following the last element.\n\n //\n\n // At this point
the insertion index is not equal to head or tail.\n\n // Also the circular array can store at least one more element.\n
\n\n //\n\n // Depending on where the given element must be inserted the preceding or the succeeding\n\n //
elements will be shifted to make room for the element to be inserted.\n\n //\n\n // In case the preceding elements
are shifted:\n\n // * if the insertion index is greater than the head (regardless of circular array form)\n\n // ->
shift the preceding elements\n\n // * otherwise, the circular array has (2) form and the insertion index is less than
tail\n\n // -> shift all elements in the back of the array\n\n // -> shift preceding elements in the front of the
array\n\n // In case the succeeding elements are shifted:\n\n // * if the insertion index is less than the tail
(regardless of circular array form)\n\n // -> shift the succeeding elements\n\n // * otherwise, the
circular array has (2) form and the insertion index is greater than head\n\n // -> shift all elements in the front
of the array\n\n // -> shift succeeding elements in the back of the array\n\n\n val internalIndex =
internalIndex(index)\n\n if (index < (size + 1) shr 1) {\n\n // closer to the first element -> shift preceding
elements\n\n val decrementedInternalIndex = decremented(internalIndex)\n\n val decrementedHead =
decremented(head)\n\n if (decrementedInternalIndex >= head) {\n\n elementData[decrementedHead]
= elementData[head] // head can be zero\n\n elementData.copyInto(elementData, head, head + 1,
decrementedInternalIndex + 1)\n\n } else { // head > tail\n\n elementData.copyInto(elementData, head -
1, head, elementData.size) // head can't be zero\n\n elementData[elementData.size - 1] = elementData[0]\n
\n elementData.copyInto(elementData,

```

```

0, 1, decrementedInternalIndex + 1)\n }\n elementData[decrementedInternalIndex] = element\n
head = decrementedHead\n } else {\n // closer to the last element -> shift succeeding elements\n
val tail = internalIndex(size)\n if (internalIndex < tail) {\n elementData.copyInto(elementData,\n
internalIndex + 1, internalIndex, tail)\n } else { // head > tail\n elementData.copyInto(elementData,\n
1, 0, tail)\n elementData[0] = elementData[elementData.size - 1]\n
elementData.copyInto(elementData, internalIndex + 1, internalIndex, elementData.size - 1)\n }\n
elementData[internalIndex] = element\n }\n size += 1\n }\n private fun\n
copyCollectionElements(internalIndex: Int, elements: Collection<E>) {\n val iterator = elements.iterator()\n
for (index in internalIndex until elementData.size) {\n
 if (!iterator.hasNext()) break\n elementData[index] = iterator.next()\n }\n for (index in 0 until\n
head) {\n if (!iterator.hasNext()) break\n elementData[index] = iterator.next()\n }\n size +=\n
elements.size\n }\n public override fun addAll(elements: Collection<E>): Boolean {\n if\n
(elements.isEmpty()) return false\n ensureCapacity(this.size + elements.size)\n
copyCollectionElements(internalIndex(size), elements)\n return true\n }\n public override fun\n
addAll(index: Int, elements: Collection<E>): Boolean {\n AbstractList.checkPositionIndex(index, size)\n
if (elements.isEmpty())\n return false\n } else if (index == size) {\n return addAll(elements)\n
}\n ensureCapacity(this.size + elements.size)\n val tail = internalIndex(size)\n val internalIndex =\n
internalIndex(index)\n val elementsSize\n
= elements.size\n if (index < (size + 1) shr 1) {\n // closer to the first element -> shift preceding\n
elements\n var shiftedHead = head - elementsSize\n if (internalIndex >= head) {\n if\n
(shiftedHead >= 0) {\n elementData.copyInto(elementData, shiftedHead, head, internalIndex)\n
}\n } else { // head < tail, insertion leads to head >= tail\n shiftedHead += elementData.size\n val\n
elementsToShift = internalIndex - head\n val shiftToBack = elementData.size - shiftedHead\n }\n
if (shiftToBack >= elementsToShift) {\n elementData.copyInto(elementData, shiftedHead, head,\n
internalIndex)\n } else {\n elementData.copyInto(elementData, shiftedHead, head, head +\n
shiftToBack)\n elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)\n
 }\n }\n } else { // head > tail, internalIndex < tail\n
elementData.copyInto(elementData, shiftedHead, head, elementData.size)\n if (elementsSize >=\n
internalIndex)\n elementData.copyInto(elementData, elementData.size - elementsSize, 0,\n
internalIndex)\n } else {\n elementData.copyInto(elementData, elementData.size -\n
elementsSize, 0, elementsSize)\n elementData.copyInto(elementData, 0, elementsSize, internalIndex)\n
 }\n }\n head = shiftedHead\n copyCollectionElements(negativeMod(internalIndex -\n
elementsSize), elements)\n } else {\n // closer to the last element -> shift succeeding elements\n
val shiftedInternalIndex = internalIndex + elementsSize\n if (internalIndex < tail) {\n if (tail +\n
elementsSize <= elementData.size) {\n elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)\n
 } else { // head <\n
tail, insertion leads to head >= tail\n if (shiftedInternalIndex >= elementData.size) {\n
elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)\n } else\n
{\n val shiftToFront = tail + elementsSize - elementData.size\n
elementData.copyInto(elementData, 0, tail - shiftToFront, tail)\n elementData.copyInto(elementData,\n
shiftedInternalIndex, internalIndex, tail - shiftToFront)\n }\n }\n } else { // head > tail,\n
internalIndex > head\n elementData.copyInto(elementData, elementsSize, 0, tail)\n if\n
(shiftedInternalIndex >= elementData.size) {\n elementData.copyInto(elementData, shiftedInternalIndex\n
- elementData.size,\n
internalIndex, elementData.size)\n } else {\n elementData.copyInto(elementData, 0,\n
elementData.size - elementsSize, elementData.size)\n elementData.copyInto(elementData,\n
shiftedInternalIndex, internalIndex, elementData.size - elementsSize)\n }\n }\n
copyCollectionElements(internalIndex, elements)\n }\n return true\n }\n public override fun

```

```

get(index: Int): E {\n AbstractList.checkElementIndex(index, size)\n\n return
internalGet(internalIndex(index))\n } \n\n public override fun set(index: Int, element: E): E {\n
AbstractList.checkElementIndex(index, size)\n\n val internalIndex = internalIndex(index)\n val oldElement
= internalGet(internalIndex)\n elementData[internalIndex] = element\n\n return oldElement\n } \n\n
public override fun contains(element: E): Boolean = indexOf(element) != -1\n\n public override
fun indexOf(element: E): Int {\n val tail = internalIndex(size)\n\n if (head < tail) {\n for (index in
head until tail) {\n if (element == elementData[index]) return index - head\n } \n } else if (head
>= tail) {\n for (index in head until elementData.size) {\n if (element == elementData[index]) return
index - head\n } \n } for (index in 0 until tail) {\n if (element == elementData[index]) return
index + elementData.size - head\n } \n\n return -1\n } \n\n public override fun
lastIndexOf(element: E): Int {\n val tail = internalIndex(size)\n\n if (head < tail) {\n for (index in tail
- 1 downTo head) {\n if (element == elementData[index]) return index - head\n } \n } else if
(head > tail) {\n for (index in tail - 1 downTo 0) {\n if (element == elementData[index])
return index + elementData.size - head\n } \n } for (index in elementData.lastIndex downTo head) {\n
 if (element == elementData[index]) return index - head\n } \n\n return -1\n } \n\n public
override fun remove(element: E): Boolean {\n val index = indexOf(element)\n if (index == -1) return
false\n removeAt(index)\n return true\n } \n\n public override fun removeAt(index: Int): E {\n
AbstractList.checkElementIndex(index, size)\n\n if (index == lastIndex) {\n return removeLast()\n }
else if (index == 0) {\n return removeFirst()\n } \n\n val internalIndex = internalIndex(index)\n
val element = internalGet(internalIndex)\n\n if (index < size shr 1) {\n // closer to the first element ->
shift preceding elements\n if (internalIndex >= head) {\n elementData.copyInto(elementData, head
+ 1, head, internalIndex)\n } else { // head > tail, internalIndex < head\n
elementData.copyInto(elementData, 1, 0, internalIndex)\n elementData[0] = elementData[elementData.size
- 1]\n elementData.copyInto(elementData, head + 1, head, elementData.size - 1)\n } \n\n
elementData[head] = null\n head = incremented(head)\n } else {\n // closer to the last element ->
shift succeeding elements\n val internalLastIndex = internalIndex(lastIndex)\n\n if (internalIndex <=
internalLastIndex) {\n elementData.copyInto(elementData, internalIndex, internalIndex + 1,
internalLastIndex + 1)\n } else { // head > tail, internalIndex > head\n
elementData.copyInto(elementData, internalIndex, internalIndex + 1, elementData.size)\n
elementData[elementData.size - 1] = elementData[0]\n elementData.copyInto(elementData,
0, 1, internalLastIndex + 1)\n } \n\n elementData[internalLastIndex] = null\n } \n\n size -= 1\n\n
return element\n } \n\n public override fun removeAll(elements: Collection<E>): Boolean = filterInPlace {
!elements.contains(it) } \n\n public override fun retainAll(elements: Collection<E>): Boolean = filterInPlace {
elements.contains(it) } \n\n private inline fun filterInPlace(predicate: (E) -> Boolean): Boolean {\n if
(this.isEmpty() || elementData.isEmpty())\n return false\n\n val tail = internalIndex(size)\n var
newTail = head\n var modified = false\n\n if (head < tail) {\n for (index in head until tail) {\n
val element = elementData[index]\n\n @SuppressWarnings("UNCHECKED_CAST")\n if
(predicate(element as E))\n elementData[newTail++] = element\n else\n modified =
true\n } \n\n elementData.fill(null, newTail, tail)\n } else {\n for (index in head until elementData.size) {\n
val element = elementData[index]\n elementData[index] = null\n\n @SuppressWarnings("UNCHECKED_CAST")\n if (predicate(element as E))\n elementData[newTail++] = element\n else\n modified = true\n } \n\n newTail =
positiveMod(newTail)\n for (index in 0 until tail) {\n val element = elementData[index]\n
elementData[index] = null\n\n @SuppressWarnings("UNCHECKED_CAST")\n if (predicate(element as
E)) {\n elementData[newTail] = element\n newTail = incremented(newTail)\n }
else {\n modified = true\n } \n } \n } \n\n if (modified)\n size =
negativeMod(newTail

```

```

- head)\n\n return modified\n }\n\n public override fun clear() {\n val tail = internalIndex(size)\n if (head < tail) {\n elementData.fill(null, head, tail)\n } else if (isEmpty()) {\n elementData.fill(null, head, elementData.size)\n elementData.fill(null, 0, tail)\n }\n head = 0\n size = 0\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override fun <T> toArray(array: Array<T>): Array<T> {\n @Suppress("UNCHECKED_CAST")\n val dest = (if (array.size >= size) array else arrayOfNulls(array, size)) as Array<Any?>\n val tail = internalIndex(size)\n if (head < tail) {\n elementData.copyInto(dest, startIndex = head, endIndex = tail)\n } else if (isEmpty()) {\n elementData.copyInto(dest, destinationOffset = 0, startIndex = head, endIndex = elementData.size)\n elementData.copyInto(dest, destinationOffset = elementData.size\n - head, startIndex = 0, endIndex = tail)\n }\n if (dest.size > size) {\n dest[size] = null // null-terminate\n }\n @Suppress("UNCHECKED_CAST")\n return dest as Array<T>\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override fun toArray(): Array<Any?> {\n return toArray(arrayOfNulls<Any?>(size))\n }\n\n // for testing\n internal fun <T> testToArray(array: Array<T>): Array<T> = toArray(array)\n internal fun testToArray(): Array<Any?> = toArray()\n\n internal companion object {\n private val emptyElementData = emptyArray<Any?>()\n private const val maxArraySize = Int.MAX_VALUE - 8\n private const val defaultMinCapacity = 10\n internal fun newCapacity(oldCapacity: Int, minCapacity: Int): Int {\n // overflow-conscious\n var newCapacity = oldCapacity + (oldCapacity shr 1)\n if (newCapacity - minCapacity < 0)\n newCapacity = minCapacity\n if (newCapacity - maxArraySize > 0)\n newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize\n return newCapacity\n }\n }\n\n // For testing only\n internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {\n val tail = internalIndex(size)\n val head = if (isEmpty() || head < tail) head else head - elementData.size\n structure(head, toArray())\n }\n}"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\n/**\n * Returns a single list of all elements from all arrays in the given array.\n * @sample samples.collections.Arrays.Transformations.flattenArray\n */\npublic fun <T> Array<out Array<out T>>.flatten(): List<T> {\n val result = ArrayList<T>(sumOf { it.size })\n for (element in this) {\n result.addAll(element)\n }\n return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this array,\n * *second* list is built from the second values of each pair from this array.\n * @sample samples.collections.Arrays.Transformations.unzipArray\n */\npublic fun <T, R> Array<out Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val listT = ArrayList<T>(size)\n val listR = ArrayList<R>(size)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n/**\n * Returns `true` if this nullable array is either null or empty.\n * @sample samples.collections.Arrays.Usage.arrayIsNullOrEmpty\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun Array<*>?.isNullOrEmpty(): Boolean {\n contract {\n fun returns(false) implies (this@isNullOrEmpty != null)\n }\n return this == null || this.isEmpty()\n}\n\n/**\n * Returns this array if it's not empty\n * or the result of calling [defaultValue] function if the array is empty.\n * @sample samples.collections.Arrays.Usage.arrayIfEmpty\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("UPPER_BOUND_CANNOT_BE_ARRAY")\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<*>, C : R =\n if (isEmpty())\n defaultValue()\n else\n this\n}\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.JvmName("contentDeepEquals")\n@kotlin.js.JsName("contentDeepEqualsImpl")\ninternal fun <T> Array<out T>?.contentDeepEqualsImpl(other: Array<out T>?): Boolean {\n if (this === other) return true\n if (this == null

```

```

|| other == null || this.size != other.size) return false\n\n for (i in indices) {\n val v1 = this[i]\n val v2 = other[i]\n\n if (v1 === v2) {\n continue\n } else if (v1 == null || v2 == null) {\n return false\n }\n\n when {\n v1 is Array<*> && v2 is Array<*> -> if (!v1.contentDeepEquals(v2)) return false\n v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false\n v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false\n v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false\n v1 is LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false\n v1 is FloatArray && v2 is FloatArray -> if (!v1.contentEquals(v2)) return false\n v1 is DoubleArray && v2 is DoubleArray -> if (!v1.contentEquals(v2)) return false\n v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false\n v1 is UShortArray && v2 is UShortArray -> if (!v1.contentEquals(v2)) return false\n v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2)) return false\n v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false\n } else -> if (v1 != v2) return false\n }\n\n return true\n}\n\n\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.JvmName("contentDeepToString")\n@kotlin.js.JsName("contentDeepToStringImpl")\ninternal fun <T> Array<out T>?.contentDeepToStringImpl(): String {\n if (this == null) return "null"\n val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2 // in order not to overflow Int.MAX_VALUE\n return buildString(length) {\n contentDeepToStringInternal(this, mutableListOf())\n }\n}\n\n@OptIn(ExperimentalUnsignedTypes::class)\nprivate fun <T> Array<out T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {\n if (this in processed) {\n result.append("[...]")\n return\n }\n processed.add(this)\n result.append("[")\n for (i in indices) {\n if (i != 0) {\n result.append(", ")\n }\n val element = this[i]\n when (element) {\n null -> result.append("null")\n is Array<*> -> element.contentDeepToStringInternal(result, processed)\n is ByteArray -> result.append(element.contentToString())\n is ShortArray -> result.append(element.contentToString())\n is IntArray -> result.append(element.contentToString())\n is LongArray -> result.append(element.contentToString())\n is FloatArray -> result.append(element.contentToString())\n is DoubleArray -> result.append(element.contentToString())\n is CharArray -> result.append(element.contentToString())\n is BooleanArray -> result.append(element.contentToString())\n is UByteArray -> result.append(element.contentToString())\n is UShortArray -> result.append(element.contentToString())\n is UIntArray -> result.append(element.contentToString())\n is ULongArray -> result.append(element.contentToString())\n else -> result.append(element.toString())\n }\n }\n result.append("]")\n processed.removeAt(processed.lastIndex)\n}"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\n/** Returns true if the brittle contains optimization is enabled. See KT-45438. */\ninternal expect fun brittleContainsOptimizationEnabled(): Boolean\n\n/** Returns true if [brittleContainsOptimizationEnabled] is true and it's safe to convert this collection to a set without changing contains method behavior. */\nprivate fun <T> Collection<T>.safeToConvertToSet() = brittleContainsOptimizationEnabled() && size > 2 && this is ArrayList\n\n/** When [brittleContainsOptimizationEnabled] is true: */\n * - Converts this [Iterable] to a set if it is not a [Collection].\n * - Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.\n * - Otherwise returns this.\n\n/** When [brittleContainsOptimizationEnabled] is false: */\n * - Converts this [Iterable] to a list if it is not a [Collection].\n * - Otherwise returns this.\n\ninternal fun <T> Iterable<T>.convertToSetForSetOperationWith(source: Iterable<T>): Collection<T> =\n when (this) {\n is Set -> this\n is Collection -> when {\n source is Collection && source.size < 2 -> this\n
```

```

else -> if (this.safeToConvertToSet()) toHashSet() else this\n }\n else -> if
(brITTLECONTAINSOPTIMIZATIONENABLED()) toHashSet() else toList()\n }\n\n/**\n * When
[brITTLECONTAINSOPTIMIZATIONENABLED] is true:\n * - Converts this [Iterable] to a set if it is not a [Collection].\n * -
Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.\n * -
Otherwise returns this.\n * When [brITTLECONTAINSOPTIMIZATIONENABLED] is false:\n * - Converts this [Iterable] to a
list if it is not a [Collection].\n * - Otherwise returns this.\n */\ninternal fun <T>
Iterable<T>.convertToSetForSetOperation(): Collection<T> =\n when (this) {\n is Set -> this\n is
Collection -> if (this.safeToConvertToSet()) toHashSet() else this\n else -> if
(brITTLECONTAINSOPTIMIZATIONENABLED()) toHashSet() else toList()\n }\n\n/**\n * Converts this sequence to a set if
[brITTLECONTAINSOPTIMIZATIONENABLED] is true,\n * otherwise
converts it to a list.\n */\ninternal fun <T> Sequence<T>.convertToSetForSetOperation(): Collection<T> =\n if
(brITTLECONTAINSOPTIMIZATIONENABLED()) toHashSet() else toList()\n\n/**\n * Converts this array to a set if
[brITTLECONTAINSOPTIMIZATIONENABLED] is true,\n * otherwise converts it to a list.\n */\ninternal fun <T>
Array<T>.convertToSetForSetOperation(): Collection<T> =\n if (brITTLECONTAINSOPTIMIZATIONENABLED())
toHashSet() else asList()", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * Data class representing a value from a
collection or sequence, along with its index in that collection or sequence.\n * \n * @property value the underlying
value.\n * @property index the index of the value in the collection or sequence.\n */\npublic data class
IndexedValue<out T>(public val
index: Int, public val value: T)\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("MapAccessorsKt")\n\npackage
kotlin.collections\n\nimport kotlin.reflect.KProperty\nimport kotlin.internal.Exact\n\n/**\n * Returns the value of
the property for the given object from this read-only map.\n * @param thisRef the object for which the value is
requested (not used).\n * @param property the metadata for the property, used to get the name of property and
lookup the value corresponding to this name in the map.\n * @return the property value.\n * \n * @throws
NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit
default (see [withDefault]).\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> Map<in
String, @Exact V>.getValue(thisRef:
Any?, property: KProperty<*>): V1 =\n @Suppress("UNCHECKED_CAST")
(getOrImplicitDefault(property.name) as V1)\n\n/**\n * Returns the value of the property for the given object from
this mutable map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property
the metadata for the property, used to get the name of property and lookup the value corresponding to this name in
the map.\n * @return the property value.\n * \n * @throws NoSuchElementException when the map doesn't contain
value for the property name and doesn't provide an implicit default (see [withDefault]).\n */\n@kotlin.jvm.JvmName("getVar")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V>
MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)\n\n/**\n * Stores the value of
the property for the given object in this mutable map.\n * @param thisRef
the object for which the value is requested (not used).\n * @param property the metadata for the property, used to
get the name of property and store the value associated with that name in the map.\n * @param value the value to
set.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> MutableMap<in String, in
V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n this.put(property.name, value)\n}\n", /*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage
kotlin.collections\n\n/**\n * Returns the value for the given key, or the implicit default value for this map.\n * By

```

default no implicit value is provided for maps and a [NoSuchElementException] is thrown.\n \* To create a map with implicit default value use [withDefault]

method.\n \* \n \* @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.\n

```
*\n@kotlin.jvm.JvmName("\ngetOrImplicitDefaultNullable\n")\n@PublishedApi\ninternal fun <K, V> Map<K, V>.getOrImplicitDefault(key: K): V {\n if (this is MapWithDefault)\n return\n this.getOrImplicitDefault(key)\n return getOrElseNullable(key, { throw NoSuchElementException("\nKey $key\nis missing in the map.") })\n}\n\n/**\n * Returns a wrapper of this read-only map, having the implicit default value\n provided with the specified function [defaultValue].\n * \n * This implicit default value is used when the original\n map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function, for\n example when properties are delegated to the map.\n * \n * When this map already has an implicit default value\n provided with a former call to [withDefault], it is being replaced by this call.\n
```

```
*\npublic fun <K, V> Map<K, V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =\n when (this) {\n is MapWithDefault -> this.map.withDefault(defaultValue)\n else -> MapWithDefaultImpl(this, defaultValue)\n }\n\n/**\n * Returns a wrapper of this mutable map, having the implicit default value provided with the specified\n function [defaultValue].\n * \n * This implicit default value is used when the original map doesn't contain a value for\n the key specified\n * and a value is obtained with [Map.getValue] function, for example when properties are\n delegated to the map.\n * \n * When this map already has an implicit default value provided with a former call to\n [withDefault], it is being replaced by this call.\n * \n *\n@kotlin.jvm.JvmName("\nwithDefaultMutable\n")\npublic fun\n<K, V> MutableMap<K, V>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =\n when (this) {\n is MutableMapWithDefault -> this.map.withDefault(defaultValue)\n else ->
```

```
MutableMapWithDefaultImpl(this,
```

```
 defaultValue)\n }\n\nprivate interface MapWithDefault<K, out V> : Map<K, V> {\n public val map: Map<K, V>\n public fun getOrImplicitDefault(key: K): V\n}\n\nprivate interface MutableMapWithDefault<K, V> :\n MutableMap<K, V>, MapWithDefault<K, V> {\n public override val map: MutableMap<K, V>\n}\n\nprivate\nclass MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) :\n MapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean = map.equals(other)\n override fun\n hashCode(): Int = map.hashCode()\n override fun toString(): String = map.toString()\n override val size: Int get()\n = map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n override fun containsKey(key: K): Boolean =\n map.containsKey(key)\n override fun containsValue(value: @UnsafeVariance V): Boolean =\n map.containsValue(value)\n override fun get(key: K): V? = map.get(key)\n override val keys: Set<K> get()\n = map.keys\n
```

```
 override val values: Collection<V> get() = map.values\n override val entries: Set<Map.Entry<K, V>> get()\n = map.entries\n}\n\nprivate class MutableMapWithDefaultImpl<K, V>(public override val map: MutableMap<K, V>, private\nval default: (key: K) -> V) : MutableMapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean =\n map.equals(other)\n override fun hashCode(): Int = map.hashCode()\n override fun toString(): String =\n map.toString()\n override val size: Int get() = map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n override fun containsKey(key: K): Boolean = map.containsKey(key)\n override fun containsValue(value:\n @UnsafeVariance V): Boolean = map.containsValue(value)\n override fun get(key: K): V? = map.get(key)\n override val keys: MutableSet<K> get() = map.keys\n override val values: MutableCollection<V> get()\n = map.values\n
```

```
 override val entries: MutableSet<MutableMap.MutableEntry<K, V>> get() = map.entries\n}\n\noverride fun\nput(key: K, value: V): V? = map.put(key, value)\n override fun remove(key: K): V? = map.remove(key)\n override fun putAll(from: Map<out K, V>) = map.putAll(from)\n override fun clear() = map.clear()\n}\n\noverride\nfun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, { default(key) })\n}\n\n"/**\n * Copyright\n 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed\n by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
```

```

* \n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nimport kotlin.random.Random\n\n/**\n * Removes a single instance of the specified element
from this\n * collection, if it is present.\n *\n * Allows to overcome type-safety restriction of `remove` that requires
to pass an element of type `E`.\n *\n
 * @return `true` if the element has been successfully removed; `false` if it was not present in the collection.\n
*\n\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out
T>.remove(element: T): Boolean =\n @Suppress("UNCHECKED_CAST") (this as
MutableCollection<T>).remove(element)\n\n/**\n * Removes all of this collection's elements that are also
contained in the specified collection.\n *\n * Allows to overcome type-safety restriction of `removeAll` that requires
to pass a collection of type `Collection<E>`.\n *\n * @return `true` if any of the specified elements was removed
from the collection, `false` if the collection was not modified.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun
<@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean
=\n @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)\n\n/**\n *
Retains only the elements in this collection
that are contained in the specified collection.\n *\n * Allows to overcome type-safety restriction of `retainAll` that
requires to pass a collection of type `Collection<E>`.\n *\n * @return `true` if any element was removed from the
collection, `false` if the collection was not modified.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun
<@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =\n
@Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).retainAll(elements)\n\n/**\n * Adds the
specified [element] to this mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
MutableCollection<in T>.plusAssign(element: T) {\n this.add(element)\n}\n\n/**\n * Adds all elements of the
given [elements] collection to this mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline operator fun
<T> MutableCollection<in T>.plusAssign(elements: Iterable<T>) {\n this.addAll(elements)\n}\n\n/**\n
 * Adds all elements of the given [elements] array to this mutable collection.\n
*\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(elements:
Array<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all elements of the given [elements] sequence to this
mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in
T>.plusAssign(elements: Sequence<T>) {\n this.addAll(elements)\n}\n\n/**\n * Removes a single instance of the
specified [element] from this mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
MutableCollection<in T>.minusAssign(element: T) {\n this.remove(element)\n}\n\n/**\n * Removes all elements
contained in the given [elements] collection from this mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic
inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Iterable<T>) {\n
 this.removeAll(elements)\n}\n\n/**\n * Removes
all elements contained in the given [elements] array from this mutable collection.\n
*\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements:
Array<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Removes all elements contained in the given [elements]
sequence from this mutable collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
MutableCollection<in T>.minusAssign(elements: Sequence<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Adds
all elements of the given [elements] collection to this [MutableCollection].\n *\n\npublic fun <T>
MutableCollection<in T>.addAll(elements: Iterable<T>): Boolean {\n when (elements) {\n is Collection ->
return addAll(elements)\n else -> {\n var result: Boolean = false\n for (item in elements)\n
if (add(item)) result = true\n return result\n }\n }\n}\n\n/**\n * Adds all elements of the given
[elements] array to this [MutableCollection].\n *\n\npublic fun <T> MutableCollection<in T>.addAll(elements:
Sequence<T>): Boolean {\n var result: Boolean = false\n for (item in elements) {\n if (add(item)) result =
true\n }\n return result\n}\n\n/**\n * Adds all elements of the given [elements] array to this
[MutableCollection].\n *\n\npublic fun <T> MutableCollection<in T>.addAll(elements: Array<out T>): Boolean {\n
return addAll(elements.asList())\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are also

```



contained in the given [elements] collection.

```

public fun <T> MutableCollection<in T>.removeAll(elements:
Iterable<T>): Boolean {
 return removeAll(elements.convertToSetForSetOperationWith(this))
}

```

\* Removes all elements from this [MutableCollection] that are also contained in the given [elements] sequence.

```

public fun <T> MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {
 val set =
elements.convertToSetForSetOperation()
 return set.isNotEmpty() && removeAll(set)
}

```

\* Removes all elements from this [MutableCollection] that are also contained in the given [elements] array.

```

public fun <T> MutableCollection<in
T>.removeAll(elements: Array<out T>): Boolean {
 return elements.isNotEmpty() &&
removeAll(elements.convertToSetForSetOperation())
}

```

\* Retains only elements of this [MutableCollection] that are contained in the given [elements] collection.

```

public fun <T>
MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {
 return
retainAll(elements.convertToSetForSetOperationWith(this))
}

```

\* Retains only elements of this [MutableCollection] that are contained in the given [elements] array.

```

public fun <T> MutableCollection<in
T>.retainAll(elements: Array<out T>): Boolean {
 if (elements.isNotEmpty())
 return
retainAll(elements.convertToSetForSetOperation())
 else
 return retainNothing()
}

```

\* Retains only elements of this [MutableCollection] that are contained in the given [elements] sequence.

```

public fun <T> MutableCollection<in T>.retainAll(elements: Sequence<T>): Boolean {
 val set =
elements.convertToSetForSetOperation()
 if (set.isNotEmpty())
 return retainAll(set)
 else
 return
retainNothing()
}

```

private fun MutableCollection<\*>.retainNothing(): Boolean {
 val result = isEmpty()
 clear()
 return result
}

\* Removes all elements from this [MutableIterable] that match the given [predicate].

```

@return `true` if any element was removed from this collection, or `false` when no elements
were removed and collection was not modified.

```

```

public fun <T> MutableIterable<T>.removeAll(predicate: (T)
-> Boolean): Boolean = filterInPlace(predicate, true)

```

\* Retains only elements of this [MutableIterable] that match the given [predicate].

```

@return `true` if any element was removed from this collection,
or `false` when all elements were retained and collection was not modified.

```

```

public fun <T>
MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)

```

private fun <T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {
 var result = false
 with(iterator()) {
 while (hasNext())
 if (predicate(next()) ==
predicateResultToRemove) {
 remove()
 result = true
 }
 }
 return
result
}

\* Removes the element at the specified [index] from this list. In Kotlin one should use the [MutableList.removeAt] function instead.

```

@Deprecated("Use removeAt(index) instead.",
ReplaceWith("removeAt(index)"), level = DeprecationLevel.ERROR)

```

```

kotlin.internal.InlineOnly
public inline
fun <T> MutableList<T>.remove(index: Int): T = removeAt(index)

```

\* Removes the first element from this mutable list

and returns that removed element, or throws [NoSuchElementException] if this list is empty.

```

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T>
MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(0)

```

\* Removes the first element from this mutable list and returns that removed element, or returns `null` if this list is empty.

```

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T>
MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)

```

\* Removes the last element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.

```

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T>
MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(lastIndex)

```

\* Removes the last element from this mutable list and returns that removed element, or returns `null` if this list is empty.

```

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T>
MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)

```

\* Removes all

```

elements from this [MutableList] that match the given [predicate].\n *\n * @return `true` if any element was
removed from this collection, or `false` when no elements were removed and collection was not modified.\n
*\npublic fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate,
true)\n/>\n *\n * Retains only elements of this [MutableList] that match the given [predicate].\n *\n * @return `true`
if any element was removed from this collection, or `false` when all elements were retained and collection was not
modified.\n *\npublic fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean =
filterInPlace(predicate, false)\n\nprivate fun
<T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {\n
if (this !is RandomAccess)\n return (this as MutableIterable<T>).filterInPlace(predicate,
predicateResultToRemove)\n var writeIndex: Int = 0\n for (readIndex in 0..lastIndex) {\n val element =
this[readIndex]\n if (predicate(element) == predicateResultToRemove)\n continue\n if (writeIndex
!= readIndex)\n this[writeIndex] = element\n writeIndex++\n }\n if (writeIndex < size) {\n for
(removeIndex in lastIndex downTo writeIndex)\n removeAt(removeIndex)\n return true\n } else {\n
return false\n }\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nprivate open class ReversedListReadonly<out T>(private val delegate: List<T>) :
AbstractList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n}\n\nprivate class ReversedList<T>(private val delegate: MutableList<T>) :
AbstractMutableList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n override fun clear() = delegate.clear()\n override fun removeAt(index:
Int): T = delegate.removeAt(reverseElementIndex(index))\n override fun set(index: Int, element: T): T =
delegate.set(reverseElementIndex(index), element)\n override fun add(index: Int, element: T) {\n
delegate.add(reversePositionIndex(index), element)\n }\n}\n\nprivate fun List<*>.reverseElementIndex(index:
Int) =\n if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element
index $index must be in range [${0..lastIndex}].")\n\nprivate fun List<*>.reversePositionIndex(index: Int) =\n if
(index in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range
[${0..size}].")\n\n/>\n *\n * Returns a reversed read-only view of the original List.\n * All changes made in the
original list will be reflected in the reversed one.\n * @sample samples.collections.ReversedViews.asReversedList\n
*\npublic fun <T> List<T>.asReversed(): List<T> = ReversedListReadonly(this)\n/>\n *\n * Returns a reversed
mutable view of the original mutable List.\n * All changes made in the original list will be reflected in the reversed
one and vice versa.\n * @sample samples.collections.ReversedViews.asReversedMutableList\n
*\n\n@kotlin.jvm.JvmName("asReversedMutable")\n\npublic fun <T> MutableList<T>.asReversed():
MutableList<T> = ReversedList(this)\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n@file:OptIn(Experimenta
lTypeInference::class)\n\npackage kotlin.sequences\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.experimental.ExperimentalTypeInference\n\n/>\n *\n * Builds a
[Sequence] lazily yielding values one by one.\n *\n * @see kotlin.sequences.generateSequence\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n *\n\n@SinceKotlin("1.3")\n\npublic fun <T>
sequence(@BuilderInference block: suspend SequenceScope<T>().-> Unit): Sequence<T> = Sequence {
iterator(block) }\n\n@SinceKotlin("1.3")\n\n@Deprecated("Use 'sequence { }' function instead.",
ReplaceWith("sequence(builderAction)"), level =
DeprecationLevel.ERROR)\n\n@kotlin.internal.InlineOnly\n\npublic

```

```

inline fun <T> buildSequence(@BuilderInference noline builderAction: suspend SequenceScope<T>().-> Unit):
Sequence<T> = Sequence { iterator(builderAction) }\n\n/**\n * Builds an [Iterator] lazily yielding values one by
one.\n * \n * @sample samples.collections.Sequences.Building.buildIterator\n * @sample
samples.collections.Iterables.Building.iterable\n * \n @SinceKotlin("1.3")\n\npublic fun <T>
iterator(@BuilderInference block: suspend SequenceScope<T>().-> Unit): Iterator<T> {\n val iterator =
SequenceBuilderIterator<T>()\n iterator.nextStep = block.createCoroutineUnintercepted(receiver = iterator,
completion = iterator)\n return iterator\n}\n\n@SinceKotlin("1.3")\n@Deprecated("Use 'iterator { }' function
instead.", ReplaceWith("iterator(builderAction)"), level =
DeprecationLevel.ERROR)\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> buildIterator(@BuilderInference
noline builderAction: suspend SequenceScope<T>().-> Unit):
Iterator<T> = iterator(builderAction)\n\n/**\n * The scope for yielding values of a [Sequence] or an [Iterator],
provides [yield] and [yieldAll] suspension functions.\n * \n * @see sequence\n * @see iterator\n * \n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*\n @RestrictsSuspension\n @SinceKotlin("1.3")\n\npublic abstract class SequenceScope<in T> internal
constructor() {\n /**\n * Yields a value to the [Iterator] being built and suspends\n * until the next value is
requested.\n * \n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n * \n public abstract suspend fun yield(value:
T)\n * \n /**\n * Yields all values from the `iterator` to the [Iterator] being built\n * and suspends until all these
values are iterated and the next one is requested.\n * \n * \n * The sequence of values returned by the given iterator can be potentially infinite.\n * \n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * \n public abstract suspend fun
yieldAll(iterator: Iterator<T>)\n * \n /**\n * Yields a collections of values to the [Iterator] being built\n * and
suspends until all these values are iterated and the next one is requested.\n * \n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * \n public suspend fun yieldAll(elements:
Iterable<T>) {\n if (elements is Collection && elements.isEmpty()) return\n return
yieldAll(elements.iterator())\n }\n * \n /**\n * Yields potentially infinite sequence of values to the [Iterator]
being built\n * and suspends until all these values are iterated and the next one is requested.\n * \n * \n * The
sequence can be potentially infinite.\n * \n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n
*\n public suspend fun yieldAll(sequence: Sequence<T>) =
yieldAll(sequence.iterator())\n}\n\n@Deprecated("Use SequenceScope class instead.",
ReplaceWith("SequenceScope<T>"), level = DeprecationLevel.ERROR)\n\npublic typealias SequenceBuilder<T> =
SequenceScope<T>\n\nprivate typealias State = Int\n\nprivate const val State_NotReady: State = 0\n\nprivate const
val State_ManyNotReady: State = 1\n\nprivate const val State_ManyReady: State = 2\n\nprivate const val State_Ready:
State = 3\n\nprivate const val State_Done: State = 4\n\nprivate const val State_Failed: State = 5\n\nprivate class
SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {\n private var state =
State_NotReady\n private var nextValue: T? = null\n private var nextIterator: Iterator<T>? = null\n var
nextStep: Continuation<Unit>? = null\n\n override fun hasNext(): Boolean {\n while (true) {\n when
(state) {\n State_NotReady -> {}\n
 State_ManyNotReady ->\n if (nextIterator!!.hasNext()) {\n state =
State_ManyReady\n return true\n } else {\n nextIterator = null\n
 }\n State_Done -> return false\n State_Ready, State_ManyReady -> return true\n
 else ->
throw exceptionalState()\n }\n state = State_Failed\n val step = nextStep!!\n nextStep =
null\n step.resume(Unit)\n }\n }\n\n override fun next(): T {\n when (state) {\n State_NotReady, State_ManyNotReady -> return nextNotReady()\n State_ManyReady -> {\n state
= State_ManyNotReady\n return nextIterator!!.next()\n }\n State_Ready -> {\n state
= State_NotReady\n @Suppress("UNCHECKED_CAST")\n val

```

```

 result = nextValue as T\n nextValue = null\n return result\n }\n else -> throw
exceptionalState()\n }\n }\n private fun nextNotReady(): T {\n if (!hasNext()) throw
NoSuchElementException() else return next()\n }\n private fun exceptionalState(): Throwable = when (state)
{\n State_Done -> NoSuchElementException()\n State_Failed -> IllegalStateException("\Iterator has
failed.\")\n else -> IllegalStateException("\Unexpected state of the iterator: $state")\n }\n override
suspend fun yield(value: T) {\n nextValue = value\n state = State_Ready\n return
suspendCoroutineUninterceptedOrReturn { c ->\n nextStep = c\n COROUTINE_SUSPENDED\n
}\n }\n override suspend fun yieldAll(iterator: Iterator<T>) {\n if (!iterator.hasNext()) return\n
nextIterator = iterator\n state = State_ManyReady\n return suspendCoroutineUninterceptedOrReturn
{ c ->\n nextStep = c\n COROUTINE_SUSPENDED\n }\n }\n // Completion continuation
implementation\n override fun resumeWith(result: Result<Unit>) {\n result.getOrThrow() // just rethrow
exception if it is there\n state = State_Done\n }\n override val context: CoroutineContext\n get() =
EmptyCoroutineContext\n }\n /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n package kotlin.collections\n internal fun checkWindowSizeStep(size: Int, step:
Int) {\n require(size > 0 && step > 0) {\n if (size != step)\n "\Both size $size and step $step must be
greater than zero.\n else\n "\size $size must be greater than zero.\n }\n }\n internal fun <T>
Sequence<T>.windowedSequence(size: Int, step:
Int, partialWindows: Boolean, reuseBuffer: Boolean): Sequence<List<T>> {\n checkWindowSizeStep(size,
step)\n return Sequence { windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }\n }\n internal
fun <T> windowedIterator(iterator: Iterator<T>, size: Int, step: Int, partialWindows: Boolean, reuseBuffer:
Boolean): Iterator<List<T>> {\n if (!iterator.hasNext()) return EmptyIterator\n return iterator<List<T>> {\n
val bufferInitialCapacity = size.coerceAtMost(1024)\n val gap = step - size\n if (gap >= 0) {\n var
buffer = ArrayList<T>(bufferInitialCapacity)\n var skip = 0\n for (e in iterator) {\n if (skip >
0) { skip -= 1; continue }\n buffer.add(e)\n if (buffer.size == size) {\n yield(buffer)\n
 if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)\n skip = gap\n }\n
 }\n if (buffer.isNotEmpty()) {\n if (partialWindows || buffer.size == size) yield(buffer)\n }\n
 } else {\n var buffer = RingBuffer<T>(bufferInitialCapacity)\n for (e in iterator) {\n
buffer.add(e)\n if (buffer.isFull()) {\n if (buffer.size < size) { buffer =
buffer.expanded(maxCapacity = size); continue }\n yield(if (reuseBuffer) buffer else
ArrayList(buffer))\n buffer.removeFirst(step)\n }\n if (partialWindows) {\n
while (buffer.size > step) {\n yield(if (reuseBuffer) buffer else ArrayList(buffer))\n
buffer.removeFirst(step)\n }\n if (buffer.isNotEmpty()) yield(buffer)\n }\n }\n }\n internal class MovingSubList<out E>(private val list: List<E>) : AbstractList<E>(), RandomAccess {\n
private
var fromIndex: Int = 0\n private var _size: Int = 0\n fun move(fromIndex: Int, toIndex: Int) {\n
checkRangeIndexes(fromIndex, toIndex, list.size)\n this.fromIndex = fromIndex\n this._size = toIndex -
fromIndex\n }\n override fun get(index: Int): E {\n checkElementIndex(index, _size)\n return
list[fromIndex + index]\n }\n override val size: Int get() = _size\n /**\n * Provides ring buffer
implementation.\n * Buffer overflow is not allowed so [add] doesn't overwrite tail but raises an exception.\n
*/\n private class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int) : AbstractList<T>(),
RandomAccess {\n init {\n require(filledSize >= 0) {\n "\ring buffer filled size should not be negative but it is
$filledSize" }\n require(filledSize <= buffer.size) {\n "\ring buffer filled size: $filledSize cannot be larger than
the buffer size: ${buffer.size}" }\n }\n constructor(capacity: Int)
: this(arrayOfNulls<Any?>(capacity), 0)\n private val capacity = buffer.size\n private var startIndex: Int =
0\n override var size: Int = filledSize\n private set\n override fun get(index: Int): T {\n
checkElementIndex(index, size)\n @Suppress("\UNCHECKED_CAST")\n return

```



```

erimentalUnsignedTypes\nprivate fun partition(\n array: UIntArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n array: UIntArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left < index - 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n// ULongArray
=====
erimentalUnsignedTypes\nprivate fun partition(\n array: ULongArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n array: ULongArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left < index - 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n// Interfaces
=====
\n/**\n * Sorts the given array using qsort algorithm.\n */\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UByteArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UShortArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UIntArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: ULongArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)"
\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\nimport kotlin.internal.InlineOnly\n\n/**\n * Compares this object with the specified object for order. Returns zero if this object is equal\n * to the specified [other] object, a negative number if it's less than [other], or a positive number\n * if it's greater than [other].\n */\n * This function delegates to [Comparable.compareTo] and allows to call it in infix form.\n */\n@InlineOnly\n@SinceKotlin("1.6")\npublic inline infix fun <T> Comparable<T>.compareTo(other: T): Int = this.compareTo(other)\n\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.contracts\nimport kotlin.internal.ContractsDsl\nimport kotlin.internal.InlineOnly\n\n/**\n * This marker distinguishes the experimental contract declaration API and is used to opt-in for that feature\n * when declaring contracts of user functions.\n */\n * Any usage of a declaration annotated with `@ExperimentalContracts` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalContracts::class)`,\n * or by using the compiler argument `-opt-in=kotlin.contracts.ExperimentalContracts`.\n */\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.3")\n@RequiresOptIn\n@MustBeDocumented\npublic annotation class ExperimentalContracts\n\n/**\n * Provides a scope, where the functions of the contract DSL, such as [returns], [callsInPlace], etc.,\n * can be used to describe the contract of a function.\n */\n * This type is used as a receiver type of the lambda function passed to the [contract] function.\n */\n * @see contract\n */\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin("1.3")\npublic interface ContractBuilder {\n\n /**\n * Describes a situation when a function returns normally, without any exceptions thrown.\n */\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n */\n * // @sample samples.contracts.returnsContract\n * @ContractsDsl public fun returns(): Returns\n\n /**\n * Describes a situation when a function returns normally with the specified return [value].\n */\n * The possible values of

```

```

[value] are limited to `true`, `false` or `null`.
 * Use [SimpleEffect.implies] function to describe a
conditional effect that happens in such case.
 * // @sample samples.contracts.returnsTrueContract
// @sample samples.contracts.returnsFalseContract
// @sample samples.contracts.returnsNullContract
@ContractsDsl public fun returns(value: Any?): Returns
 /**
 * Describes a situation when a function
returns normally with any value that is not `null`.
 * Use [SimpleEffect.implies] function
to describe a conditional effect that happens in such case.
 * // @sample
samples.contracts.returnsNotNullContract
 @ContractsDsl public fun returnsNotNull(): ReturnsNotNull
 /**
 * Specifies that the function parameter [lambda] is invoked in place.
 * This contract specifies
that:
 * 1. the function [lambda] can only be invoked during the call of the owner function,
 * and it won't be
invoked after that owner function call is completed;
 * 2. _(optionally)_ the function [lambda] is invoked the
amount of times specified by the [kind] parameter,
 * see the [InvocationKind] enum for possible values.
 * A function declaring the `callsInPlace` effect must be _inline_.
 * // @sample
samples.contracts.callsInPlaceAtMostOnceContract
 * @sample
samples.contracts.callsInPlaceAtLeastOnceContract
 * @sample
samples.contracts.callsInPlaceExactlyOnceContract
 * @sample
samples.contracts.callsInPlaceUnknownContract
 * // @ContractsDsl public fun <R> callsInPlace(lambda: Function<R>, kind: InvocationKind =
InvocationKind.UNKNOWN): CallsInPlace
 /**
 * Specifies how many times a function invokes its function
parameter in place.
 * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.
 * // @ContractsDsl
@ExperimentalContracts
@SinceKotlin("1.3")
public enum class InvocationKind {
 /**
 * A function parameter will be invoked one time or not invoked at all.
 * // @sample
samples.contracts.callsInPlaceAtMostOnceContract
 @ContractsDsl AT_MOST_ONCE,
 /**
 * A
function parameter will be invoked one or more times.
 * // @sample
samples.contracts.callsInPlaceAtLeastOnceContract
 @ContractsDsl AT_LEAST_ONCE,
 /**
 * A
function parameter will be invoked exactly one time.
 * // @sample
samples.contracts.callsInPlaceExactlyOnceContract
 @ContractsDsl EXACTLY_ONCE,
 /**
 * A function parameter is called in place, but it's unknown how
many times it can be called.
 * // @sample samples.contracts.callsInPlaceUnknownContract
@ContractsDsl UNKNOWN
}
 /**
 * Specifies the contract of a function.
 * The contract description
must be at the beginning of a function and have at least one effect.
 * Only the top-level functions can have a
contract for now.
 * @param builder the lambda where the contract of a function is described with the help of
the [ContractBuilder] members.
 * // @sample samples.contracts.returnsContract
 * @sample
samples.contracts.returnsTrueContract
 * @sample samples.contracts.returnsFalseContract
 * @sample
samples.contracts.returnsNullContract
 * @sample samples.contracts.returnsNotNullContract
 * @sample
samples.contracts.callsInPlaceAtMostOnceContract
 * @sample
samples.contracts.callsInPlaceAtLeastOnceContract
 * @sample
samples.contracts.callsInPlaceExactlyOnceContract
 * @sample
samples.contracts.callsInPlaceUnknownContract
 * // @ContractsDsl
@ExperimentalContracts
@InlineOnly
@SinceKotlin("1.3")
@Suppress("UNUSED_PARAMETER")
public inline fun contract(builder:
ContractBuilder.() -> Unit) { }
 /**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
 * // @package kotlin.coroutines
 /**
 * Marks coroutine context element that
intercepts coroutine continuations.
 * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the
interceptor and
 * intercepts all coroutine continuations with [interceptContinuation] invocations.
 * // @sample
[ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that
 * its implementation delegates [get][CoroutineContext.Element.get] and
[minusKey][CoroutineContext.Element.minusKey]

```

```

* to [getPolymorphicElement] and [minusPolymorphicKey] respectively.\n * [ContinuationInterceptor] subtypes
can be extracted from the coroutine context using either [ContinuationInterceptor.Key]\n * or subtype key if it
extends [AbstractCoroutineContextKey].\n *^\n@SinceKotlin("1.3")\npublic interface ContinuationInterceptor :
CoroutineContext.Element {\n /**\n * The key that defines *the* context interceptor.\n */\n companion
object Key : CoroutineContext.Key<ContinuationInterceptor>\n\n /**\n * Returns continuation that wraps the
original [continuation], thus intercepting all resumptions.\n * This function is invoked by coroutines framework
when needed and the resulting continuations are\n * cached internally per each instance of the original
[continuation].\n */\n * This function may simply return original [continuation] if it does not want to intercept
this particular continuation.\n */\n * When the original [continuation]
completes, coroutine framework invokes [releaseInterceptedContinuation]\n * with the resulting continuation if it
was intercepted, that is if `interceptContinuation` had previously\n * returned a different continuation instance.\n
*/\n public fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>\n\n /**\n *
Invoked for the continuation instance returned by [interceptContinuation] when the original\n * continuation
completes and will not be used anymore. This function is invoked only if [interceptContinuation]\n * had returned
a different continuation instance from the one it was invoked with.\n */\n * Default implementation does
nothing.\n */\n * @param continuation Continuation instance returned by this interceptor's
[interceptContinuation] invocation.\n */\n public fun releaseInterceptedContinuation(continuation:
Continuation<*>) {\n /** do nothing by default */\n }\n\n public override operator
fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>): E? {\n // getPolymorphicKey
specialized for ContinuationInterceptor key\n @OptIn(ExperimentalStdlibApi::class)\n if (key is
AbstractCoroutineContextKey<*, *>) {\n @Suppress("UNCHECKED_CAST")\n return if
(key.isSubKey(this.key)) key.tryCast(this) as? E else null\n }\n @Suppress("UNCHECKED_CAST")\n
return if (ContinuationInterceptor === key) this as E else null\n }\n\n public override fun minusKey(key:
CoroutineContext.Key<*>): CoroutineContext {\n // minusPolymorphicKey specialized for
ContinuationInterceptor key\n @OptIn(ExperimentalStdlibApi::class)\n if (key is
AbstractCoroutineContextKey<*, *>) {\n return if (key.isSubKey(this.key) && key.tryCast(this) != null)
EmptyCoroutineContext else this\n }\n return if (ContinuationInterceptor === key)
EmptyCoroutineContext else this\n }\n}\n", "/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.coroutines\n\n/**\n * Persistent context for the coroutine. It is an indexed set of [Element] instances.\n * An
indexed set is a mix between a set and a map.\n * Every element in this set has a unique [Key].\n
*/\n@SinceKotlin("1.3")\npublic interface CoroutineContext {\n /**\n * Returns the element with the given
[key] from this context or `null`.\n */\n public operator fun <E : Element> get(key: Key<E>): E?\n\n /**\n
* Accumulates entries of this context starting with [initial] value and applying [operation]\n * from left to right to
current accumulator value and each element of this context.\n */\n public fun <R> fold(initial: R, operation: (R,
Element) -> R): R\n\n /**\n * Returns a context containing elements from this
context and elements from other [context].\n * The elements from this context with the same key as in the other
one are dropped.\n */\n public operator fun plus(context: CoroutineContext): CoroutineContext =\n if
(context === EmptyCoroutineContext) this else // fast path -- avoid lambda creation\n context.fold(this) {
acc, element ->\n val removed = acc.minusKey(element.key)\n if (removed ===
EmptyCoroutineContext) element else {\n // make sure interceptor is always last in the context (and thus
is fast to get when present)\n val interceptor = removed[ContinuationInterceptor]\n if
(interceptor == null) CombinedContext(removed, element) else {\n val left =
removed.minusKey(ContinuationInterceptor)\n if (left === EmptyCoroutineContext)
CombinedContext(element, interceptor) else\n CombinedContext(CombinedContext(left,
element), interceptor)\n }\n }\n }\n\n /**\n * Returns a context containing elements
from this context, but without an element with\n * the specified [key].\n */\n public fun minusKey(key:

```



```

Key<*>: CoroutineContext {
 /**
 * Key for the elements of [CoroutineContext]. [E] is a type of element
 * with this key.
 */
 public interface Key<E : Element> {
 /**
 * An element of the [CoroutineContext].
 * An element of the coroutine context is a singleton context by itself.
 */
 public interface Element {
 CoroutineContext {
 /**
 * A key of this coroutine context element.
 */
 public val key:
 Key<*>
 }
 public override operator fun <E : Element> get(key: Key<E>): E? =
 @Suppress("UNCHECKED_CAST")
 if (this.key == key) this as E else null
 public override fun
 <R> fold(initial: R, operation: (R, Element) ->
 R): R =
 operation(initial, this)
 public override fun minusKey(key: Key<*>): CoroutineContext =
 if (this.key == key) EmptyCoroutineContext else this
 }
 }
}

/**
 * Copyright 2010-2020 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 */
package kotlin.coroutines
import
 kotlin.coroutines.CoroutineContext.Element
import
 kotlin.coroutines.CoroutineContext.Key

/**
 * Base class
 * for [CoroutineContext.Element] implementations.
 */
@SinceKotlin("1.3")
public abstract class
 AbstractCoroutineContextElement(public override val key: Key<*>) : Element {
 /**
 * Base class for
 * [CoroutineContext.Key] associated with polymorphic [CoroutineContext.Element]
 * implementation.
 * Polymorphic element implementation implies delegating its [get][Element.get] and
 * [minusKey][Element.minusKey] to [getPolymorphicElement]
 * and [minusPolymorphicKey] respectively.
 * Polymorphic elements can be extracted from the coroutine
 * context using both element key and its supertype key.
 * Example of polymorphic elements:
 */
 open class
 BaseElement : CoroutineContext.Element {
 /**
 * companion object Key : CoroutineContext.Key<BaseElement>
 * override val key: CoroutineContext.Key<*> get() = Key
 * // It is important to use getPolymorphicKey and
 * minusPolymorphicKey
 * override fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>):
 * E? = getPolymorphicElement(key)
 * override fun minusKey(key: CoroutineContext.Key<*>):
 * CoroutineContext = minusPolymorphicKey(key)
 */
 companion object Key : AbstractCoroutineContextKey<BaseElement, DerivedElement>(
 BaseElement, { it as?
 DerivedElement }) {
 // Now it is possible to query both `BaseElement` and `DerivedElement`
 someContext[BaseElement] // Returns BaseElement?,
 non-null both for BaseElement and DerivedElement instances
 someContext[DerivedElement] // Returns
 DerivedElement?, non-null only for DerivedElement instance
 }
 @param B base class of a polymorphic
 element
 @param baseKey an instance of base key
 @param E element type associated with the current key
 @param safeCast a function that can safely cast abstract [CoroutineContext.Element]
 to the concrete [E] type
 and return the element if it is a subtype of [E] or `null` otherwise.
 }

 @SinceKotlin("1.3")
 @ExperimentalStdlibApi
 public abstract class AbstractCoroutineContextKey<B :
 Element, E : B> {
 baseKey: Key,
 private val safeCast: (element: Element) -> E?
 } : Key<E> {
 private val topmostKey: Key<*> = if (baseKey is AbstractCoroutineContextKey<*, *>)
 baseKey.topmostKey else
 baseKey
 internal fun tryCast(element: Element): E? = safeCast(element)
 internal fun isSubKey(key:
 Key<*>): Boolean = key === this || topmostKey
 === key
 }

 /**
 * Returns the current element if it is associated with the given [key] in a polymorphic
 * manner
 * or `null` otherwise.
 * This method returns non-null value if either [Element.key] is equal to the given [key]
 * or if
 * the [key] is associated
 * with [Element.key] via [AbstractCoroutineContextKey].
 * See
 * [AbstractCoroutineContextKey] for the example of usage.
 */
 @SinceKotlin("1.3")
 @ExperimentalStdlibApi
 public fun <E : Element>
 Element.getPolymorphicElement(key: Key<E>): E? {
 if (key is AbstractCoroutineContextKey<*, *>) {
 @Suppress("UNCHECKED_CAST")
 return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
 }
 @Suppress("UNCHECKED_CAST")
 return if (this.key === key) this as E else null
 }

 /**
 * Returns empty coroutine context if the element is associated with the given [key]
 * in a polymorphic manner
 * or `null` otherwise.
 * This method returns empty context if either [Element.key] is equal to the

```

given [key] or if the [key] is associated\n \* with [Element.key] via [AbstractCoroutineContextKey].\n \* See [AbstractCoroutineContextKey] for the example of usage.\n

```

\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic fun Element.minusPolymorphicKey(key: Key<>):
CoroutineContext {\n if (key is AbstractCoroutineContextKey<*, *>) {\n return if (key.isSubKey(this.key)
&& key.tryCast(this) != null) EmptyCoroutineContext else this\n }\n return if (this.key === key)
EmptyCoroutineContext else this\n}\n\n/*\n * An empty coroutine context.\n */\n@SinceKotlin("1.3")\npublic
object EmptyCoroutineContext : CoroutineContext, Serializable {\n private const val serialVersionUID: Long =
0\n private fun readResolve(): Any = EmptyCoroutineContext\n\n public override fun <E : Element> get(key:
Key<E>): E? = null\n public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial\n public
override fun plus(context: CoroutineContext): CoroutineContext
= context\n public override fun minusKey(key: Key<*>): CoroutineContext = this\n public override fun
hashCode(): Int = 0\n public override fun toString(): String = "EmptyCoroutineContext"\n}\n\n//-----
- internal impl -----\n\n// this class is not exposed, but is hidden inside implementations\n// this is a left-
biased list, so that `plus` works naturally\n@SinceKotlin("1.3")\ninternal class CombinedContext(\n private val
left: CoroutineContext,\n private val element: Element\n) : CoroutineContext, Serializable {\n\n override fun <E
: Element> get(key: Key<E>): E? {\n var cur = this\n while (true) {\n cur.element[key]?.let { return
it }\n val next = cur.left\n if (next is CombinedContext) {\n cur = next\n } else {\n
return next[key]\n }\n }\n }\n\n public override fun <R> fold(initial: R, operation: (R, Element)
-> R):
R =\n operation(left.fold(initial, operation), element)\n\n public override fun minusKey(key: Key<*>):
CoroutineContext {\n element[key]?.let { return left }\n val newLeft = left.minusKey(key)\n return
when {\n newLeft === left -> this\n newLeft === EmptyCoroutineContext -> element\n else ->
CombinedContext(newLeft, element)\n }\n }\n\n private fun size(): Int {\n var cur = this\n var size
= 2\n while (true) {\n cur = cur.left as? CombinedContext ?: return size\n size++\n }\n }\n\n private fun contains(element: Element): Boolean =\n get(element.key) == element\n\n private fun
containsAll(context: CombinedContext): Boolean {\n var cur = context\n while (true) {\n if
(!contains(cur.element)) return false\n val next = cur.left\n if (next is CombinedContext) {\n
cur = next\n }\n }\n }\n\n else {\n return contains(next as Element)\n }\n}\n\n override fun equals(other: Any?):
Boolean =\n this === other || other is CombinedContext && other.size() == size() &&
other.containsAll(this)\n\n override fun hashCode(): Int = left.hashCode() + element.hashCode()\n\n override
fun toString(): String =\n "[" + fold("") { acc, element ->\n if (acc.isEmpty()) element.toString() else
"\$acc, \$element"\n } + "]"

private fun writeReplace(): Any {\n val n = size()\n val elements =
arrayOfNulls<CoroutineContext>(n)\n var index = 0\n fold(Unit) { _, element -> elements[index++] =
element }\n check(index == n)\n @Suppress("UNCHECKED_CAST")\n return Serialized(elements
as Array<CoroutineContext>)\n}\n\n private class Serialized(val elements: Array<CoroutineContext>) :
Serializable {\n companion object {\n private const val serialVersionUID:
Long = 0L\n }\n\n private fun readResolve(): Any = elements.fold(EmptyCoroutineContext,
CoroutineContext::plus)\n }\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("IntrinsicsKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
kotlin.coroutines.intrinsics\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.*\nimport
kotlin.internal.InlineOnly\n\n/*\n * Obtains the current continuation instance inside suspend functions and either
suspends\n * currently running coroutine or returns result immediately without suspension.\n */\n * If the [block]
returns the special [COROUTINE_SUSPENDED] value, it means that suspend function did suspend the execution
and will\n * not return any result immediately. In this case, the [Continuation] provided to the [block] shall

```

be resumed by invoking [Continuation.resumeWith] at some moment in the future when the result becomes available to resume the computation. Otherwise, the return value of the [block] must have a type assignable to [T] and represents the result of this suspend function. It means that the execution was not suspended and the [Continuation] provided to the [block] shall not be invoked. As the result type of the [block] is declared as `Any?` and cannot be correctly type-checked, its proper return type remains on the conscience of the suspend function's author. Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's thread without going through the [ContinuationInterceptor] that might be present in the coroutine's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is established. [Continuation.intercepted] can be used to acquire the intercepted continuation. Note that it is not recommended to call either [Continuation.resume] nor [Continuation.resumeWithException] functions synchronously in the same stackframe where suspension function is run. Use [suspendCoroutine] as a safer way to obtain current continuation instance.

```
*\n@SinceKotlin("1.3")\n@InlineOnly\n@Suppress("UNUSED_PARAMETER",
"RedundantSuspendModifier")\npublic suspend inline fun <T>
suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {\n contract {
callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n throw NotImplementedError("Implementation of
suspendCoroutineUninterceptedOrReturn is intrinsic")\n}\n\n/**\n * This value is used as a return value of
[suspendCoroutineUninterceptedOrReturn] `block` argument to state that\n * the execution was suspended and will
not return any result immediately.\n * **Note: this value should not be used in general code.** Using it outside
of the context of\n * `suspendCoroutineUninterceptedOrReturn`\n * function return value (including, but not limited to,\n * storing this value in other properties, returning it from other
functions, etc)\n * can lead to unspecified behavior of the code.\n */\n// It is implemented as property with getter to
avoid ProGuard <clint> problem with multifile IntrinsicKt class\n@SinceKotlin("1.3")\npublic val
COROUTINE_SUSPENDED: Any get() = CoroutineSingletons.COROUTINE_SUSPENDED\n\n// Using enum
here ensures two important properties:\n// 1. It makes SafeContinuation serializable with all kinds of serialization
frameworks (since all of them natively support enums)\n// 2. It improves debugging experience, since you clearly
see toString() value of those objects and what package they come from\n@SinceKotlin("1.3")\n@PublishedApi //
This class is Published API via serialized representation of SafeContinuation, don't rename/move\ninternal enum
class CoroutineSingletons { COROUTINE_SUSPENDED, UNDECIDED, RESUMED }\n", "/*\n * Copyright
2010-2018
```

JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. \n\npackage kotlin.experimental\n\n/\*\*  
Performs a bitwise AND operation between the two values.

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.and(other: Byte): Byte =
(this.toInt() and other.toInt()).toByte()\n\n/** Performs a bitwise OR operation between the two values.
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.or(other: Byte): Byte =
(this.toInt() or other.toInt()).toByte()\n\n/** Performs a bitwise XOR operation between the two values.
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.xor(other: Byte): Byte =
(this.toInt() xor other.toInt()).toByte()\n\n/** Inverts the bits in this value.
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.inv(): Byte =
(this.toInt().inv()).toByte()\n\n/**
```

Performs a bitwise AND operation between the two values.

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.and(other: Short): Short =
(this.toInt() and other.toInt()).toShort()\n\n/** Performs a bitwise OR operation between the two values.
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short =
(this.toInt() or other.toInt()).toShort()\n\n/** Performs a bitwise XOR operation between the two values.
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short =
(this.toInt() xor other.toInt()).toShort()\n\n/** Inverts the bits in this value.
```

```

*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short =
(this.toInt().inv()).toShort()\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache
2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/**\n * The
experimental marker for type inference augmenting annotations.\n *\n * Any usage of a declaration annotated with
`@ExperimentalTypeInference` must be accepted either by\n * annotating that usage with the [OptIn] annotation,
e.g. `@OptIn(ExperimentalTypeInference::class)`,\n * or by using the compiler argument `opt-
in=kotlin.experimental.ExperimentalTypeInference`.\n */\n\n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(A
nnotationTarget.ANNOTATION_CLASS)\n@SinceKotlin("1.3")\npublic annotation class
ExperimentalTypeInference\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding
type should be ignored during type inference.\n */\n\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
NoInfer\n\n/**\n * Specifies that the constraint built for the type during type inference should be an equality one.\n */\n\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
Exact\n\n/**\n * Specifies that a corresponding member has the lowest priority in overload resolution.\n */\n\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,
AnnotationTarget.CONSTRUCTOR)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
LowPriorityInOverloadResolution\n\n/**\n * Specifies that the corresponding member has the highest priority in
overload resolution. Effectively this means that\n * an extension annotated with this annotation will win in overload
resolution over a member with the same signature.\n */\n\n@Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal
annotation class HidesMembers\n\n/**\n * The value of this type parameter should be mentioned in input types
(argument types, receiver type or expected type).\n */\n\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal
annotation class OnlyInputTypes\n\n/**\n * Specifies that this function should not be called directly without
inlining\n */\n\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,
AnnotationTarget.PROPERTY_GETTER,
AnnotationTarget.PROPERTY_SETTER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
InlineOnly\n\n/**\n * Specifies that this declaration can have dynamic receiver type.\n */\n\n@Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
DynamicExtension\n\n/**\n * The value of this parameter should be a property reference expression (`this::foo`),
referencing a `lateinit` property,\n * the backing
field of which is accessible at the point where the corresponding argument is passed.\n */\n\n@Target(AnnotationTarget.VALUE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\n@SinceK
otlin("1.2")\ninternal annotation class AccessibleLateinitPropertyLiteral\n\n/**\n * Specifies that this declaration is
only completely supported since the specified version.\n *\n * The Kotlin compiler of an earlier version is going to
report a diagnostic on usages of this declaration.\n * The diagnostic message can be specified with [message], or via
[errorCode] (takes less space, but might not be immediately clear\n * to the user). The diagnostic severity can be
specified with [level]: WARNING/ERROR mean that either a warning or an error\n * is going to be reported,
HIDDEN means that the declaration is going to be removed\n from resolution completely.\n *\n * [versionKind]
specifies which version should be compared with the [version] value, when compiling the usage of the annotated
declaration.\n */\n

```

Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE\_VERSION] was supported, so the Kotlin compiler before 1.2 is going to\n \* treat any [RequireKotlin] as if it requires the language version. Since 1.2, the

```

Kotlin compiler supports\n * [RequireKotlinVersionKind.LANGUAGE_VERSION],
[RequireKotlinVersionKind.COMPILER_VERSION] and [RequireKotlinVersionKind.API_VERSION].\n * If the
actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),\n *
Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a
requirement\n * on the language version.\n *\n * This annotation is erased at compile time; its arguments are stored
in a more compact form in the Kotlin metadata.\n *\n @Target(AnnotationTarget.CLASS,
AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR,
AnnotationTarget.TYPEALIAS)\n @Retention(AnnotationRetention.SOURCE)\n @Repeatable\n @SinceKotlin("1.
2")\n internal
annotation class RequireKotlin(\n val version: String,\n val message: String = "",\n val level:
DeprecationLevel = DeprecationLevel.ERROR,\n val versionKind: RequireKotlinVersionKind =
RequireKotlinVersionKind.LANGUAGE_VERSION,\n val errorCode: Int = -1)\n\n /**\n * The kind of the
version that is required by [RequireKotlin].\n *\n @SinceKotlin("1.2")\n internal enum class
RequireKotlinVersionKind {\n LANGUAGE_VERSION,\n COMPILER_VERSION,\n
API_VERSION,\n }\n\n /**\n * Specifies that this declaration is a part of special DSL, used for constructing
function's contract.\n *\n @Retention(AnnotationRetention.BINARY)\n @SinceKotlin("1.2")\n internal annotation
class ContractsDsl\n\n /**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\n package kotlin.properties\n\n import kotlin.reflect.KProperty\n\n /**\n *
Standard property delegates.\n *\n public object Delegates {\n /**\n * Returns a property delegate for a
read/write property with a non-`null` value that is initialized not during\n * object construction time but at a later
time. Trying to read the property before the initial value has been\n * assigned results in an exception.\n *
*\n * @sample samples.properties.Delegates.notNullDelegate\n *
*\n public fun <T : Any> notNull():
ReadWriteProperty<Any?, T> = NotNullVar()\n /**\n * Returns a property delegate for a read/write property
that calls a specified callback function when changed.\n *
*\n @param initialValue the initial value of the property.\n
*\n @param onChange the callback which is called after the change of the property is made. The value of the
property\n * has already been changed when this callback is invoked.\n *
*\n @sample
samples.properties.Delegates.observableDelegate\n *\n public
inline fun <T> observable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue:
T) -> Unit):\n ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n
\n override fun afterChange(property: KProperty<>, oldValue: T, newValue: T) = onChange(property, oldValue,
newValue)\n }\n }\n /**\n * Returns a property delegate for a read/write property that calls a specified
callback function when changed,\n *
*\n allowing the callback to veto the modification.\n *
*\n @param initialValue the initial value of the property.\n
*\n @param onChange the callback which is called before a change to the
property value is attempted.\n *
*\n The value of the property hasn't been changed yet, when this callback is
invoked.\n *
*\n If the callback returns `true` the value of the property is being set to the new value,\n *
*\n and if the
callback returns `false` the new value is discarded and the property remains
its old value.\n *
*\n @sample samples.properties.Delegates.vetoableDelegate\n *\n @sample
samples.properties.Delegates.throwVetoableDelegate\n *\n public inline fun <T> vetoable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):\n
ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n override fun
beforeChange(property: KProperty<*>, oldValue: T, newValue: T): Boolean = onChange(property, oldValue,
newValue)\n }\n }\n\n private class NotNullVar<T : Any>(): ReadWriteProperty<Any?, T> {\n private var
value: T? = null\n public override fun getValue(thisRef: Any?, property: KProperty<*>): T {\n return value
?: throw IllegalStateException("Property ${property.name} should be initialized before get.")\n }\n public
override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {\n this.value = value\n
}\n }\n\n /**\n

```

```

* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Base interface that can be used for implementing
property delegates of read-only properties.\n *\n * This is provided only for convenience; you don't have to extend
this interface\n * as long as your property delegate has methods with the same signatures.\n *\n * @param T the
type of object which owns the delegated property.\n * @param V the type of the property value.\n */\n\npublic fun
interface ReadOnlyProperty<in T, out V> {\n /**\n * Returns the value of the property for the given object.\n
* @param thisRef the object for which the value is requested.\n * @param property the metadata for the
property.\n * @return the property value.\n */\n public operator fun getValue(thisRef:
T, property: KProperty<*>): V\n}\n\n/**\n * Base interface that can be used for implementing property delegates of
read-write properties.\n *\n * This is provided only for convenience; you don't have to extend this interface\n *
as long as your property delegate has methods with the same signatures.\n *\n * @param T the type of object which
owns the delegated property.\n * @param V the type of the property value.\n */\n\npublic interface
ReadWriteProperty<in T, V> : ReadOnlyProperty<T, V> {\n /**\n * Returns the value of the property for the
given object.\n * @param thisRef the object for which the value is requested.\n * @param property the
metadata for the property.\n * @return the property value.\n */\n public override operator fun
getValue(thisRef: T, property: KProperty<*>): V\n\n /**\n * Sets the value of the property for the given
object.\n * @param thisRef the object for which the value is requested.\n * @param property the metadata
for the property.\n * @param value the value to set.\n */\n public operator fun setValue(thisRef: T,
property: KProperty<*>, value: V)\n}\n\n/**\n * Base interface that can be used for implementing property delegate
providers.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as long as your
delegate provider has a method with the same signature.\n *\n * @param T the type of object which owns the
delegated property.\n * @param D the type of property delegates this provider provides.\n */\n\n@SinceKotlin("1.4")\n\npublic fun interface PropertyDelegateProvider<in T, out D> {\n /**\n * Returns the
delegate of the property for the given object.\n *\n * This function can be used to extend the logic of creating
the object (e.g. perform validation checks)\n * to which the property implementation is delegated.\n *\n *
@param thisRef the object for which property delegate is requested.\n * @param property the metadata
for the property.\n * @return the property delegate.\n */\n public operator fun provideDelegate(thisRef: T,
property: KProperty<*>): D\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Implements the core logic of a property delegate for a read/write property that calls callback functions when
changed.\n * @param initialValue the initial value of the property.\n */\n\npublic abstract class
ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {\n private var value = initialValue\n\n /**\n * The callback which is called before a change to the property value is attempted.\n * The value of the
property hasn't been changed yet, when this callback is invoked.\n * If the callback returns `true` the value of
the property is being set to the new value,\n * and if the callback returns `false` the new value is discarded and
the property remains its old value.\n */\n protected open fun beforeChange(property: KProperty<*>, oldValue:
V, newValue: V): Boolean = true\n\n /**\n * The callback which is called after the change of the property is
made. The value of the property\n * has already been changed when this callback is invoked.\n */\n protected open fun afterChange(property: KProperty<*>, oldValue: V, newValue: V): Unit {}\n\n public override fun
getValue(thisRef: Any?, property: KProperty<*>): V {\n return value\n }\n\n public override fun
setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n val oldValue = this.value\n if
(!beforeChange(property, oldValue, value)) {\n return\n }\n this.value = value\n afterChange(property, oldValue, value)\n }\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress("PackageDirectoryMismatch")\n\npackage
kotlin\n\nimport kotlin.reflect.*\n\n/**\n * An extension operator that allows delegating a read-only property of type

```

```

[V]\n * to a property reference to a property of type [V] or its subtype.\n *\n * @receiver A property reference to a
read-only or mutable property of type [V] or its subtype.\n * The reference is without a receiver, i.e. it either
references a top-level property or\n * has the receiver bound to it.\n *\n * Example:\n *\n * ```\n * class Login(val
username: String)\n * val defaultLogin = Login("Admin")\n * val defaultUsername by defaultLogin::username\n *
// equivalent to\n * val defaultUserName get() = defaultLogin.username\n * ```\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V>
KProperty0<V>.getValue(thisRef:
Any?, property: KProperty<*>): V {\n return get()\n}\n\n/**\n * An extension operator that allows delegating a
mutable property of type [V]\n * to a property reference to a mutable property of the same type [V].\n *\n *
@receiver A property reference to a mutable property of type [V].\n * The reference is without a receiver, i.e. it
either references a top-level property or\n * has the receiver bound to it.\n *\n * Example:\n *\n * ```\n * class
Login(val username: String, var incorrectAttemptCounter: Int = 0)\n * val defaultLogin = Login("Admin")\n * var
defaultLoginAttempts by defaultLogin::incorrectAttemptCounter\n * // equivalent to\n * var defaultLoginAttempts:
Int\n * get() = defaultLogin.incorrectAttemptCounter\n * set(value) { defaultLogin.incorrectAttemptCounter =
value }\n * ```\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V>
KMutableProperty0<V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n set(value)\n}\n\n/**\n * An extension operator that allows delegating a read-only member or extension property of type [V]\n * to a
property reference to a member or extension property of type [V] or its subtype.\n *\n * @receiver A property
reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound receiver of
type [T].\n *\n * Example:\n *\n * ```\n * class Login(val username: String)\n * val Login.user by
Login::username\n * // equivalent to\n * val Login.user get() = this.username\n * ```\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KProperty1<T,
V>.getValue(thisRef: T, property: KProperty<*>): V {\n return get(thisRef)\n}\n\n/**\n * An extension operator
that allows delegating a mutable member or extension property of type [V]\n * to a property reference to a member
or extension mutable property of the same type [V].\n *\n * @receiver A property reference to a read-only or
mutable property
of type [V] or its subtype.\n * The reference has an unbound receiver of type [T].\n *\n * Example:\n *\n * ```\n *
class Login(val username: String, var incorrectAttemptCounter: Int)\n * var Login.attempts by
Login::incorrectAttemptCounter\n * // equivalent to\n * var Login.attempts: Int\n * get() =
this.incorrectAttemptCounter\n * set(value) { this.incorrectAttemptCounter = value }\n * ```\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KMutableProperty1<T,
V>.setValue(thisRef: T, property: KProperty<*>, value: V) {\n set(thisRef, value)\n}\n\n"/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@package kotlin.random\n\nimport
kotlin.math.nextDown\n\n/**\n * An abstract class that is implemented by random number generator algorithms.\n
*\n * The companion object [Random.Default]
is the default instance of [Random].\n *\n * To get a seeded instance of random generator use [Random] function.\n
*\n * @sample samples.random.Randoms.defaultRandom\n *\n@SinceKotlin("1.3")\npublic abstract class
Random {\n\n /**\n * Gets the next random [bitCount] number of bits.\n *\n * Generates an `Int` whose
lower [bitCount] bits are filled with random values and the remaining upper bits are zero.\n *\n * @param
bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.\n *\n *
@sample samples.random.Randoms.nextBits\n *\n public abstract fun nextBits(bitCount: Int): Int\n\n /**\n * Gets the next random `Int` from the random number generator.\n *\n * Generates an `Int` random value
uniformly distributed between `Int.MIN_VALUE` and `Int.MAX_VALUE` (inclusive).\n *\n * @sample
samples.random.Randoms.nextInt\n *\n public open fun nextInt(): Int = nextBits(32)\n\n /**\n * Gets the next random non-negative `Int` from the random number generator less than the specified
[until] bound.\n *\n * Generates an `Int` random value uniformly distributed between `0` (inclusive) and the
specified [until] bound (exclusive).\n *\n * @param until must be positive.\n *\n * @throws

```

```

IllegalArgumentException if [until] is negative or zero.\n *\n * @sample
samples.random.Randoms.nextIntFromUntil\n *^\n public open fun nextInt(until: Int): Int = nextInt(0, until)\n\n
/**\n * Gets the next random `Int` from the random number generator in the specified range.\n *\n *
Generates an `Int` random value uniformly distributed between the specified [from] (inclusive) and [until]
(exclusive) bounds.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n
*\n * @sample samples.random.Randoms.nextIntFromUntil\n *^\n public open fun nextInt(from: Int, until:
Int): Int {\n checkRangeBounds(from, until)\n val n = until - from\n if (n > 0 || n == Int.MIN_VALUE)
{\n val rnd = if (n and -n == n) {\n val bitCount = fastLog2(n)\n nextBits(bitCount)\n
 } else {\n var v: Int\n do {\n val bits = nextInt().ushr(1)\n v = bits % n\n
 } while (bits - v + (n - 1) < 0)\n v\n }\n return from + rnd\n } else {\n while
(true) {\n val rnd = nextInt()\n if (rnd in from until until) return rnd\n }\n }\n }\n\n
/**\n * Gets the next random `Long` from the random number generator.\n *\n * Generates a `Long` random
value uniformly distributed between `Long.MIN_VALUE` and `Long.MAX_VALUE` (inclusive).\n *\n *
@param until\n * @throws IllegalArgumentException if [until] is negative or zero.\n *\n * @sample
samples.random.Randoms.nextLong\n *^\n public open fun nextLong():
Long = nextInt().toLong().shl(32) + nextInt()\n\n
/**\n * Gets the next random non-negative `Long` from the
random number generator less than the specified [until] bound.\n *\n * Generates a `Long` random value
uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n *\n *
@param until\n * must be positive.\n *\n * @throws IllegalArgumentException if [until] is negative or zero.\n
*\n * @sample samples.random.Randoms.nextLongFromUntil\n *^\n public open fun nextLong(until: Long): Long =
nextInt(0, until)\n\n
/**\n * Gets the next random `Long` from the random number generator in the specified
range.\n *\n * Generates a `Long` random value uniformly distributed between the specified [from] (inclusive)
and [until] (exclusive) bounds.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to
[until].\n *\n * @sample samples.random.Randoms.nextLongFromUntil\n
 *^\n public open fun nextLong(from: Long, until: Long): Long {\n checkRangeBounds(from, until)\n
val n = until - from\n if (n > 0) {\n val rnd: Long\n if (n and -n == n) {\n val nLow =
n.toInt()\n val nHigh = (n ushr 32).toInt()\n rnd = when {\n nLow != 0 -> {\n
 val bitCount = fastLog2(nLow)\n // toUInt().toLong()\n
 nextBits(bitCount).toLong() and 0xFFFF_FFFF\n }\n nHigh == 1 -> {\n
 //
 toUInt().toLong()\n nextInt().toLong() and 0xFFFF_FFFF\n } else -> {\n val
bitCount = fastLog2(nHigh)\n nextBits(bitCount).toLong().shl(32) + (nextInt().toLong() and
0xFFFF_FFFF)\n }\n }\n } else {\n var v: Long\n
 do {\n val bits = nextLong().ushr(1)\n v = bits % n\n } while (bits - v + (n - 1)
< 0)\n rnd = v\n }\n return from + rnd\n } else {\n while (true) {\n val
rnd = nextLong()\n if (rnd in from until until) return rnd\n }\n }\n }\n }\n\n
/**\n * Gets the
next random [Boolean] value.\n *\n * @sample samples.random.Randoms.nextBoolean\n *^\n public open
fun nextBoolean(): Boolean = nextBits(1) != 0\n\n
/**\n * Gets the next random [Double] value uniformly
distributed between 0 (inclusive) and 1 (exclusive).\n *\n * @sample samples.random.Randoms.nextDouble\n
 *^\n public open fun nextDouble(): Double = doubleFromParts(nextBits(26), nextBits(27))\n\n
/**\n * Gets
the next random non-negative `Double` from the random number generator less than the specified [until] bound.\n
*\n
*\n * Generates a `Double` random value uniformly distributed between 0 (inclusive) and [until] (exclusive).\n
*\n * @throws IllegalArgumentException if [until] is negative or zero.\n *\n * @sample
samples.random.Randoms.nextDoubleFromUntil\n *^\n public open fun nextDouble(until: Double): Double =
nextDouble(0.0, until)\n\n
/**\n * Gets the next random `Double` from the random number generator in the
specified range.\n *\n * Generates a `Double` random value uniformly distributed between the specified [from]
(inclusive) and [until] (exclusive) bounds.\n *\n * [from] and [until] must be finite otherwise the behavior is
unspecified.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n *\n
*\n * @sample samples.random.Randoms.nextDoubleFromUntil\n *^\n public open fun nextDouble(from: Double,

```



```

until: Double): Double {
 checkRangeBounds(from, until)
 val size = until - from
 val r = if (size.isInfinite() && from.isFinite() && until.isFinite()) {
 val r1 = nextDouble() * (until / 2 -
from / 2)
 from + r1 + r1
 } else {
 from + nextDouble() * size
 }
 return if (r >=
until) until.nextDown() else r
}

/**
 * Gets the next random [Float] value uniformly distributed
between 0 (inclusive) and 1 (exclusive).
 */
@sample samples.random.Randoms.nextFloat()

public open fun nextFloat(): Float = nextBits(24) / (1 shl 24).toFloat()

/**
 * Fills a subrange of the
specified byte [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive
 * with random
bytes.
 */
@return [array] with the subrange filled with random bytes.
 */
@sample
samples.random.Randoms.nextBytes()

public open fun nextBytes(array: ByteArray, fromIndex: Int = 0,
toIndex: Int = array.size): ByteArray {
 require(fromIndex in
0..array.size && toIndex in 0..array.size) {
 "fromIndex ($fromIndex) or toIndex ($toIndex) are out of range:
0..${array.size}."
 }
 require(fromIndex <= toIndex) {
 "fromIndex ($fromIndex) must be not greater than
toIndex ($toIndex)."
 }
 val steps = (toIndex - fromIndex) / 4
 var position = fromIndex
 repeat(steps) {
 val v = nextInt()
 array[position] = v.toByte()
 array[position + 1] =
v.ushr(8).toByte()
 array[position + 2] = v.ushr(16).toByte()
 array[position + 3] =
v.ushr(24).toByte()
 position += 4
 }
 val remainder = toIndex - position
 val vr =
nextInt(remainder * 8)
 for (i in 0 until remainder) {
 array[position + i] = vr.ushr(i * 8).toByte()
 }
 return array
}

/**
 * Fills the specified byte [array] with random bytes and returns it.
 */
@return [array] filled with random bytes.

 */
@sample samples.random.Randoms.nextBytes()

public open fun nextBytes(array: ByteArray):
ByteArray = nextBytes(array, 0, array.size)

/**
 * Creates a byte array of the specified [size], filled with
random bytes.
 */
@sample samples.random.Randoms.nextBytes()

public open fun
nextBytes(size: Int): ByteArray = nextBytes(ByteArray(size))

/**
 * The default random number
generator.
 */
@sample samples.random.Randoms.defaultRandom()

companion object Default : Random(),
Serializable {
 private val defaultRandom: Random = defaultPlatformRandom()
 private object
Serialized : Serializable {
 private const val serialVersionUID = 0L
 private fun readResolve():
Any = Random()
 }
 private fun writeReplace(): Any = Serialized

 override fun nextBits(bitCount: Int): Int = defaultRandom.nextBits(bitCount)
 override fun nextInt(): Int =
defaultRandom.nextInt()
 override fun nextInt(until: Int): Int = defaultRandom.nextInt(until)
 override
fun nextInt(from: Int, until: Int): Int = defaultRandom.nextInt(from, until)
 override fun nextLong(): Long =
defaultRandom.nextLong()
 override fun nextLong(until: Long): Long = defaultRandom.nextLong(until)
 override
fun nextLong(from: Long, until: Long): Long = defaultRandom.nextLong(from, until)
 override fun
nextBoolean(): Boolean = defaultRandom.nextBoolean()
 override fun nextDouble(): Double =
defaultRandom.nextDouble()
 override fun nextDouble(until: Double): Double =
defaultRandom.nextDouble(until)
 override fun nextDouble(from: Double, until: Double): Double =
defaultRandom.nextDouble(from, until)
 override fun nextFloat(): Float = defaultRandom.nextFloat()
}

override
fun nextBytes(array: ByteArray): ByteArray = defaultRandom.nextBytes(array)
override fun nextBytes(size:
Int): ByteArray = defaultRandom.nextBytes(size)
override fun nextBytes(array: ByteArray, fromIndex: Int,
toIndex: Int): ByteArray =
defaultRandom.nextBytes(array, fromIndex, toIndex)
}

/**
 * Returns a repeatable random number generator seeded with the given [seed] `Int` value.
 */
@sample samples.random.Randoms.seededRandom()

Two generators with
the same seed produce the same sequence of values within the same version of Kotlin runtime.
 */
Note:
Future versions of Kotlin may change the algorithm of this seeded number generator so that it will return
 * a
sequence of values different from the current one for a given seed.
 */
@sample
On JVM the returned generator is NOT
thread-safe. Do not invoke it from multiple threads without proper synchronization.
 */
@sample
samples.random.Randoms.seededRandom()

@SinceKotlin("1.3")
public fun Random(seed: Int): Random =

```

```

XorWowRandom(seed, seed.shr(31))\n\n**\n * Returns a repeatable random number generator seeded with the
given [seed] `Long` value.\n *\n * Two generators with the same seed produce the same sequence of values within
the same version of Kotlin runtime.\n *\n * *Note:* Future versions of Kotlin may change the algorithm of this
seeded number generator so that it will return\n * a sequence of values different from the current one for a given
seed.\n *\n * On JVM the returned generator is NOT thread-safe. Do not invoke it from multiple threads without
proper synchronization.\n *\n * @sample samples.random.Randoms.seededRandom\n
*\n@SinceKotlin("1.3")\npublic fun Random(seed: Long): Random = XorWowRandom(seed.toInt(),
seed.shr(32).toInt())\n\n**\n * Gets the next random `Int` from the random number generator in the specified
[range].\n *\n * Generates an `Int` random value uniformly distributed in the specified [range]:\n * from `range.start`
inclusive to `range.endInclusive` inclusive.\n
*\n * @throws IllegalArgumentException if [range] is empty.\n *\n@SinceKotlin("1.3")\npublic fun
Random.nextInt(range: IntRange): Int = when {\n range.isEmpty() -> throw IllegalArgumentException("Cannot
get random in empty range: $range")\n range.last < Int.MAX_VALUE -> nextInt(range.first, range.last + 1)\n
range.first > Int.MIN_VALUE -> nextInt(range.first - 1, range.last) + 1\n else -> nextInt()\n}\n\n**\n * Gets the
next random `Long` from the random number generator in the specified [range].\n *\n * Generates a `Long` random
value uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive`
inclusive.\n *\n * @throws IllegalArgumentException if [range] is empty.\n *\n@SinceKotlin("1.3")\npublic fun
Random.nextLong(range: LongRange): Long = when {\n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n range.last < Long.MAX_VALUE ->
nextLong(range.first, range.last
+ 1)\n range.first > Long.MIN_VALUE -> nextLong(range.first - 1, range.last) + 1\n else ->
nextLong()\n}\n\ninternal expect fun defaultPlatformRandom(): Random\ninternal expect fun
doubleFromParts(hi26: Int, low27: Int): Double\n\ninternal fun fastLog2(value: Int): Int = 31 -
value.countLeadingZeroBits()\n\n**\n Takes upper [bitCount] bits (0..32) from this number. *\n\ninternal fun
Int.takeUpperBits(bitCount: Int): Int =\n this.ushr(32 - bitCount) and (-bitCount).shr(31)\n\ninternal fun
checkRangeBounds(from: Int, until: Int) = require(until > from) { boundsErrorMessage(from, until) }\n\ninternal fun
checkRangeBounds(from: Long, until: Long) = require(until > from) { boundsErrorMessage(from, until) }\n\ninternal fun
checkRangeBounds(from: Double, until: Double) = require(until > from) { boundsErrorMessage(from, until)
}\n\ninternal fun boundsErrorMessage(from: Any, until: Any) = "Random range is empty: [$from,
$until)."\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n@npackage kotlin.random\n\n**\n * Gets the next random [UInt]
from the random number generator.\n *\n * Generates a [UInt] random value uniformly distributed between
[UInt.MIN_VALUE] and [UInt.MAX_VALUE] (inclusive).\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(): UInt = nextInt().toUInt()\n\n**\n * Gets the next random [UInt] from the random number
generator less than the specified [until] bound.\n *\n * Generates a [UInt] random value uniformly distributed
between `0` (inclusive) and the specified [until] bound (exclusive).\n *\n * @throws IllegalArgumentException if
[until] is zero.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(until: UInt): UInt = nextUInt(0u, until)\n\n**\n * Gets the next random
[UInt] from the random number generator in the specified range.\n *\n * Generates a [UInt] random value
uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * @throws
IllegalArgumentException if [from] is greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(from: UInt, until: UInt): UInt {\n checkUIntRangeBounds(from, until)\n\n val signedFrom =
from.toInt() xor Int.MIN_VALUE\n val signedUntil = until.toInt() xor Int.MIN_VALUE\n\n val signedResult =
nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE\n return signedResult.toUInt()\n}\n\n**\n * Gets the next
random [UInt] from the random number generator in the specified [range].\n *\n * Generates a [UInt] random value

```

```

uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n
*\n * @throws IllegalArgumentException if [range] is
empty.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextInt(range: UIntRange): UInt = when {\n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n range.last < UInt.MAX_VALUE ->
nextInt(range.first, range.last + 1u)\n range.first > UInt.MIN_VALUE -> nextInt(range.first - 1u, range.last) +
1u\n else -> nextInt()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator.\n *\n * Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and
[ULong.MAX_VALUE] (inclusive).\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(): ULong = nextLong().toULong()\n\n/**\n * Gets the next random [ULong] from the random
number generator less than the specified [until] bound.\n *\n * Generates a [ULong] random value uniformly
distributed between `0` (inclusive) and the specified [until] bound
(exclusive).\n *\n * @throws IllegalArgumentException if [until] is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(until: ULong): ULong = nextULong(0uL, until)\n\n/**\n * Gets the next random [ULong] from
the random number generator in the specified range.\n *\n * Generates a [ULong] random value uniformly
distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * @throws
IllegalArgumentException if [from] is greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(from: ULong, until: ULong): ULong {\n checkULongRangeBounds(from, until)\n\n val
signedFrom = from.toLong() xor Long.MIN_VALUE\n val signedUntil = until.toLong() xor
Long.MIN_VALUE\n\n val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE\n
return signedResult.toULong()\n}\n\n/**\n * Gets the next random
[ULong] from the random number generator in the specified [range].\n *\n * Generates a [ULong] random value
uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n
*\n * @throws IllegalArgumentException if [range] is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(range: ULongRange): ULong = when {\n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n range.last < ULong.MAX_VALUE -
> nextULong(range.first, range.last + 1u)\n range.first > ULong.MIN_VALUE -> nextULong(range.first - 1u,
range.last) + 1u\n else -> nextULong()\n}\n\n/**\n * Fills the specified unsigned byte [array] with random bytes
and returns it.\n *\n * @return [array] filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray):
UByteArray {\n nextBytes(array.asByteArray())\n
return array\n}\n\n/**\n * Creates an unsigned byte array of the specified [size], filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(size: Int): UByteArray
= nextBytes(size).asUByteArray()\n\n/**\n * Fills a subrange of the specified `UByte` [array] starting from
[fromIndex] inclusive and ending [toIndex] exclusive with random UBytes.\n *\n * @return [array] with the
subrange filled with random bytes.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Random.nextUBytes(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {\n
nextBytes(array.asByteArray(), fromIndex, toIndex)\n return array\n}\n\n\ninternal fun
checkUIntRangeBounds(from: UInt, until: UInt) = require(until > from) { boundsErrorMessage(from, until)
}\ninternal fun checkULongRangeBounds(from: ULong, until: ULong) = require(until > from) {
boundsErrorMessage(from, until) }\n", "/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@n\npackage kotlin.random\n\n/**\n *
Random number generator, using Marsaglia's "xorwow" algorithm\n *\n * Cycles after 2^192 - 2^32 repetitions.\n

```

\* For more details, see Marsaglia, George (July 2003). "Xorshift RNGs". Journal of Statistical Software. 8 (14). doi:10.18637/jss.v008.i14 Available at <https://www.jstatsoft.org/v08/i14/paper>

```

internal class
XorWowRandom internal constructor(
 private var x: Int,
 private var y: Int,
 private var z: Int,
 private
var w: Int,
 private var v: Int,
 private var addend: Int) : Random(), Serializable {
 internal
constructor(seed1: Int, seed2: Int) :
 this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr
4))
 init {
 require((x or y or z or w or v) != 0) { "Initial
state must have at least one non-zero element." }
 // some trivial seeds can produce several values with
zeroes in upper bits, so we discard first 64
 repeat(64) { nextInt() }
 override fun nextInt(): Int {
 // Equivalent to the xorxow algorithm
 // From Marsaglia, G. 2003. Xorshift RNGs. J. Statis. Soft. 8, 14, p.
5
 var t = x
 t = t xor (t ushr 2)
 x = y
 y = z
 z = w
 val v0 = v
 w = v0
 t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)
 v = t
 addend += 362437
 return t + addend
 }
 override fun nextBits(bitCount: Int): Int =
 nextInt().takeUpperBits(bitCount)
 private companion object {
 private const val serialVersionUID: Long = 0L
 }
 /* Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.
Use of this source code is governed by the Apache 2.0 license that
can be found in
the license/LICENSE.txt file.
*/
}
}

```

@file:kotlin.jvm.JvmMultifileClass @file:kotlin.jvm.JvmName("RangesKt") \n \n package  
kotlin.ranges \n \n /\*\* Represents a range of [Comparable] values. \n \n private open class ComparableRange<T :  
Comparable<T>>(\n override val start: T,\n override val endInclusive: T) : ClosedRange<T> {\n override  
fun equals(other: Any?): Boolean {\n return other is ComparableRange<\*> && (isEmpty() && other.isEmpty())  
||\n start == other.start && endInclusive == other.endInclusive)\n }\n override fun hashCode(): Int  
{\n return if (isEmpty()) -1 else 31 \* start.hashCode() + endInclusive.hashCode()\n }\n override fun  
toString(): String = "\$start..\$endInclusive"\n \n \n /\*\* Creates a range from this [Comparable] value to the  
specified [that] value. \n \n \* This value needs to be smaller than or equal to [that] value, otherwise the returned  
range will be empty. \n \n @sample samples.ranges.Ranges.rangeFromComparable\n  
\*/\n public operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this,  
that)\n \n \n /\*\* Represents a range of floating point numbers. \n \n \* Extends [ClosedRange] interface providing  
custom operation [lessThanOrEquals] for comparing values of range domain type. \n \n \* This interface is  
implemented by floating point ranges returned by [Float.rangeTo] and [Double.rangeTo] operators to \n \n \* achieve  
IEEE-754 comparison order instead of total order of floating point numbers. \n \n @SinceKotlin("1.1")\n public  
interface ClosedFloatingPointRange<T : Comparable<T>> : ClosedRange<T> {\n override fun contains(value: T):  
Boolean = lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)\n override fun isEmpty():  
Boolean = !lessThanOrEquals(start, endInclusive)\n \n /\*\* \n \n \* Compares two values of range domain type and  
returns true if first is less than or equal to second. \n \n \* \n \n fun lessThanOrEquals(a: T, b: T):  
Boolean\n }\n \n \n /\*\* \n \n \* A closed range of values of type `Double`. \n \n \* Numbers are compared with the ends of this range according to  
IEEE-754. \n \n private class ClosedDoubleRange(\n start: Double,\n endInclusive: Double) :  
ClosedFloatingPointRange<Double> {\n private val \_start = start\n private val \_endInclusive = endInclusive\n  
override val start: Double get() = \_start\n override val endInclusive: Double get() = \_endInclusive\n  
override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n  
override fun contains(value: Double): Boolean = value >= \_start && value <= \_endInclusive\n  
override fun isEmpty(): Boolean = !(\_start <= \_endInclusive)\n  
override fun equals(other: Any?): Boolean {\n return other is ClosedDoubleRange && (isEmpty() &&  
other.isEmpty()) ||\n \_start == other.\_start && \_endInclusive == other.\_endInclusive)\n }\n  
override fun hashCode(): Int {\n return if (isEmpty()) -1 else 31 \* \_start.hashCode()  
+ \_endInclusive.hashCode()\n }\n  
override fun toString(): String = "\$\_start..\$\_endInclusive"\n \n \n /\*\* \n \n \* Creates a range from this [Double] value to the specified [that] value. \n \n \* Numbers are compared with the ends  
of this range according to IEEE-754. \n \n @sample samples.ranges.Ranges.rangeFromDouble\n  
\*/\n @SinceKotlin("1.1")\n public operator fun Double.rangeTo(that: Double): ClosedFloatingPointRange<Double>

```

= ClosedDoubleRange(this, that)\n\n/**\n * A closed range of values of type `Float`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n *\nprivate class ClosedFloatRange(\n start: Float,\n endInclusive: Float)\n : ClosedFloatingPointRange<Float> {\n private val _start = start\n private val _endInclusive = endInclusive\n override val start: Float get() = _start\n override val endInclusive: Float get() = _endInclusive\n\n override fun lessThanOrEquals(a: Float, b: Float): Boolean = a <= b\n\n override fun contains(value: Float): Boolean = value >= _start && value <= _endInclusive\n\n override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n override fun equals(other: Any?): Boolean {\n return other is ClosedFloatRange && (isEmpty() && other.isEmpty() ||\n _start == other._start && _endInclusive == other._endInclusive)\n }\n\n override fun hashCode(): Int {\n return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()\n }\n\n override fun toString(): String = "$_start..$_endInclusive"\n}\n\n/**\n * Creates a range from this [Float] value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n *\n @sample samples.ranges.Ranges.rangeFromFloat\n *\n @SinceKotlin("1.1")\npublic operator fun Float.rangeTo(that: Float): ClosedFloatingPointRange<Float> = ClosedFloatRange(this, that)\n\n/**\n * Returns `true` if this iterable range contains the specified [element].\n *\n * Always returns `false` if the [element] is `null`.\n\n @SinceKotlin("1.3")\n @kotlin.internal.InlineOnly\npublic inline operator fun <T, R> R.contains(element: T?): Boolean where T : Any, R : Iterable<T>, R : ClosedRange<T> =\n element != null && contains(element)\n\ninternal fun checkStepIsPositive(isPositive: Boolean, step: Number) {\n if (!isPositive) throw IllegalArgumentException("Step must be positive, was: $step.")\n}\n\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n @file:kotlin.jvm.JvmName("KClasses")\n @file:Suppress("UNCHECKED_CAST")\n\npackage kotlin.reflect\n\nimport kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Casts the given [value] to the class represented by this [KClass] object.\n * Throws an exception if the value is `null` or if it is not an instance of this class.\n *\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.safeCast]\n\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @LowPriorityInOverloadResolution\nfun <T : Any> KClass<T>.cast(value: Any?): T {\n if (!isInstance(value)) throw ClassCastException("Value cannot be cast to $qualifiedOrSimpleName")\n return value as T\n}\n\n// TODO: replace with qualifiedName when it is fully supported in K/JS\ninternal expect val KClass<*>.qualifiedOrSimpleName: String?\n\n/**\n * Casts the given [value] to the class represented by this [KClass] object.\n * Returns `null` if the value is `null` or if it is not an instance of this class.\n *\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.cast]\n\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @LowPriorityInOverloadResolution\nfun <T : Any> KClass<T>.safeCast(value: Any?): T? {\n return if (isInstance(value)) value as T else null\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.reflect\n\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmStatic\n\n/**\n * Represents a type projection. Type projection is usually the argument to another type in a type usage.\n * For example, in the type `Array<out Number>`, `out Number` is the covariant projection of the type represented by the class `Number`.\n *\n * Type projection is either the star projection, or an entity consisting of a specific type plus optional variance.\n *\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#type-projections)\n * for more information.\n *\n @SinceKotlin("1.1")\npublic data class KTypeProjection constructor(\n /**\n * The use-site variance specified in the projection, or `null` if this is a star projection.\n *\n public val variance: KVariance?,\n /**\n * The type specified in the

```

```

projection, or `null` if this is a star projection.
 */
 public val type: KType?() {
 init {
 require((variance == null) == (type == null)) {
 if (variance == null)
 "\"Star projection must have no type specified.\"
 else
 "\"The projection variance $variance requires type to be specified.\"
 }
 }
 override fun toString(): String = when (variance) {
 null -> "\"*\n KVariance.INVARIANT
-> type.toString()\n KVariance.IN -> \"in $type\"\n KVariance.OUT -> \"out $type\"\n }
 public companion object {
 // provided for compiler access
 @JvmField
 @PublishedApi
 internal val
 star: KTypeProjection = KTypeProjection(null, null)
 /**
 * Star projection, denoted by the `*` character.
 * For example, in the type `KClass<*>`, `*` is the star projection.
 * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#star-projections) for more information.
 */
 public val STAR: KTypeProjection get() = star
 /**
 * Creates an invariant projection of a given type. Invariant projection is just the type itself,
 * without any use-site variance modifiers applied to it.
 * For example, in the type `Set<String>`, `String` is an invariant projection of the type represented by the class `String`.
 */
 @JvmStatic
 public fun invariant(type: KType): KTypeProjection =
 KTypeProjection(KVariance.INVARIANT, type)
 /**
 * Creates a contravariant projection of a given type, denoted by the `in` modifier applied to a type.
 * For example, in the type `MutableList<in Number>`, `in Number` is a contravariant projection of the type of class `Number`.
 */
 @JvmStatic
 public fun contravariant(type: KType): KTypeProjection =
 KTypeProjection(KVariance.IN, type)
 /**
 * Creates a covariant projection of a given type, denoted by the `out` modifier applied to a type.
 * For example, in the type `Array<out Number>`, `out Number` is a covariant projection of the type of class `Number`.
 */
 @JvmStatic
 public fun covariant(type: KType): KTypeProjection =
 KTypeProjection(KVariance.OUT, type)
 }
 }
}
/**
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.reflect
/**
 * Represents variance applied to a type parameter on the declaration site (*declaration-site variance*),
 * or to a type in a projection (*use-site variance*).
 */
enum class KVariance {
 /**
 * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#variance) for more information.
 */
 @see [KTypeParameter.variance]
 @see [KTypeProjection]
 @SinceKotlin("1.1")
 INvariant,
 /**
 * The affected type parameter or type is *invariant*, which means it has no variance applied to it.
 */
 INVARIANT,
 /**
 * The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.
 */
 IN,
 /**
 * The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.
 */
 OUT,
}
/**
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.reflect
/**
 * Returns a runtime representation of the given reified type [T] as an instance of [KType].
 * Note that on JVM, the created type has no annotations ([KType.annotations] returns an empty list)
 * even if the type in the source code is annotated. Support for type annotations might be added in a future version.
 */
@SinceKotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <reified T> typeOf(): KType =
 throw UnsupportedOperationException("This function is implemented as an intrinsic on all supported platforms.")
/**
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.jvm
package kotlin.jvm
/**
 * An object to which char sequences and values can be appended.
 */
expect interface Appendable {
 /**
 * Appends the specified character [value] to this Appendable and returns this instance.
 */
 @param value the character to append
 fun append(value: Char): Appendable
 /**
 * Appends the specified character sequence [value] to this

```

```

Appendable and returns this instance.\n * \n * @param value the character sequence to append. If [value] is
`null`, then the four characters `"\n\n\n\n"` are appended to this Appendable.\n * \n fun append(value:
CharSequence?): Appendable\n\n /**\n * Appends a subsequence of the specified character sequence [value] to
this Appendable and returns this instance.\n * \n * @param value the character sequence from which a
subsequence is appended. If [value] is `null`,\n * then characters are appended as if [value] contained the four
characters `"\n\n\n\n"`. \n * @param startIndex the beginning (inclusive) of the subsequence to append.\n
* @param endIndex the end (exclusive) of the subsequence to append.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.\n * \n fun append(value: CharSequence?,
startIndex: Int, endIndex: Int): Appendable\n\n\n/**\n * Appends a subsequence of the specified character
sequence [value] to this Appendable and returns this instance.\n * \n * @param value the character sequence from
which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n
* @param endIndex the end (exclusive) of the subsequence to append.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.\n
\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
fun <T : Appendable> T.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): T {\n
@Suppress("UNCHECKED_CAST")\n return append(value, startIndex, endIndex) as T\n}\n\n/**\n * Appends
all arguments to the given [Appendable].\n * \n public fun <T : Appendable> T.append(vararg value:
CharSequence?): T {\n for (item in value)\n append(item)\n return this\n}\n\n/**\n * Appends a line feed
character (`\n`) to this Appendable. *\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
Appendable.appendLine(): Appendable = append("\n")\n\n/**\n * Appends value to the given Appendable and a line
feed character (`\n`) after it. *\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
Appendable.appendLine(value: CharSequence?): Appendable = append(value).appendLine()\n\n/**\n * Appends value
to the given Appendable and a line feed character (`\n`) after it.
\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(value:
Char): Appendable = append(value).appendLine()\n\n\ninternal fun <T> Appendable.appendElement(element: T,
transform: ((T) -> CharSequence)?) {\n when {\n transform != null -> append(transform(element))\n
element is CharSequence? -> append(element)\n element is Char -> append(element)\n else ->
append(element.toString())\n } \n}\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\n\n/**\n * Trims leading whitespace characters followed by [marginPrefix] from every line of a source
string and removes\n * the first and the last lines if they are blank (notice difference blank vs empty).\n * \n * Doesn't
affect a line if it doesn't contain [marginPrefix] except the first and
the last blank lines.\n * \n * Doesn't preserve the original line endings.\n * \n * @param marginPrefix non-blank
string, which is used as a margin delimiter. Default is `|` (pipe character).\n * \n * @sample
samples.text.Strings.trimMargin\n * @see trimIndent\n * @see kotlin.text.isWhitespace\n * \n public fun
String.trimMargin(marginPrefix: String = "|"): String =\n replaceIndentByMargin("|", marginPrefix)\n\n\n/**\n *
Detects indent by [marginPrefix] as it does [trimMargin] and replace it with [newIndent].\n * \n * @param
marginPrefix non-blank string, which is used as a margin delimiter. Default is `|` (pipe character).\n * \n public
fun String.replaceIndentByMargin(newIndent: String = "|", marginPrefix: String = "|"): String {\n
require(marginPrefix.isNotBlank()) { "marginPrefix must be non-blank string." }\n val lines = lines()\n\n return
lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), { line ->\n val
firstNonWhitespaceIndex
= line.indexOfFirst { !it.isWhitespace() }\n\n when {\n firstNonWhitespaceIndex == -1 -> null\n
line.startsWith(marginPrefix, firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex +

```

```

marginPrefix.length)\n else -> null\n }\n }\n)\n\n/**\n * Detects a common minimal indent of all the
input lines, removes it from every line and also removes the first and the last\n * lines if they are blank (notice
difference blank vs empty).\n *\n * Note that blank lines do not affect the detected indent level.\n *\n * In case if
there are non-blank lines with no leading whitespace characters (no indent at all) then the\n * common indent is 0,
and therefore this function doesn't change the indentation.\n *\n * Doesn't preserve the original line endings.\n *\n *
@sample samples.text.Strings.trimIndent\n * @see trimMargin\n * @see kotlin.text.isBlank\n */\npublic fun
String.trimIndent(): String = replaceIndent("\\n")\n\n/**\n * Detects a common
minimal indent like it does [trimIndent] and replaces it with the specified [newIndent].\n */\npublic fun
String.replaceIndent(newIndent: String = "\\n"): String {\n val lines = lines()\n val minCommonIndent = lines\n
.filter(String::isNotBlank)\n .map(String::indentWidth)\n .minOrNull() ?: 0\n return
lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), { line ->
line.drop(minCommonIndent) })\n}\n\n/**\n * Prepends [indent] to every line of the original string.\n *\n * Doesn't
preserve the original line endings.\n */\npublic fun String.prependIndent(indent: String = " "): String =\n
lineSequence()\n .map {\n when {\n it.isBlank() -> {\n when {\n
it.length < indent.length -> indent\n else -> it\n }\n }\n else -> indent +
it\n }\n }\n .joinToString("\\n")\n\nprivate
fun String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }\n\nprivate fun
getIndentFunction(indent: String) = when {\n indent.isEmpty() -> { line: String -> line }\n else -> { line: String -
> indent + line }\n}\n\nprivate inline fun List<String>.reindent(\n resultSizeEstimate: Int,\n indentAddFunction:
(String) -> String,\n indentCutFunction: (String) -> String?): String {\n val lastIndex = lastIndex\n return
mapIndexedNotNull { index, value ->\n if ((index == 0 || index == lastIndex) && value.isBlank())\n null\n else\n indentCutFunction(value)?.let(indentAddFunction) ?: value\n }\n
.joinTo(StringBuilder(resultSizeEstimate), "\\n")\n .toString()\n}\n\n", /*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt
file.\n */\n\npackage kotlin.text\n\n/**\n * Defines names for Unicode symbols used in proper Typography.\n */\npublic object Typography {\n /** The character " \u2013 quotation mark */\n public const val quote:
Char = "\u0022"\n /** The character $ \u2013 dollar sign */\n public const val dollar: Char = "\u0024"\n
 /** The character & \u2013 ampersand */\n public const val amp: Char = "\u0026"\n /** The character
< \u2013 less-than sign */\n public const val less: Char = "\u003C"\n /** The character > \u2013
greater-than sign */\n public const val greater: Char = "\u003E"\n /** The non-breaking space character */\n
 public const val nbsp: Char = "\u00A0"\n /** The character × */\n public const val times: Char =
"\u00D7"\n /** The character ¢ */\n public const val cent: Char = "\u00A2"\n /** The character £
*/\n public const val pound: Char = "\u00A3"\n /** The character §
*/\n public const val section: Char = "\u00A7"\n /** The character © */\n public const val copyright:
Char = "\u00A9"\n /** The character « */\n @SinceKotlin("1.6")\n public const val leftGuillemet:
Char = "\u00AB"\n /** The character » */\n @SinceKotlin("1.6")\n public const val rightGuillemet:
Char = "\u00BB"\n /** The character ® */\n public const val registered: Char = "\u00AE"\n /** The
character ° */\n public const val degree: Char = "\u00B0"\n /** The character ± */\n public const
val plusMinus: Char = "\u00B1"\n /** The character ¶ */\n public const val paragraph: Char = "\u00B6"\n
 /** The character · */\n public const val middleDot: Char = "\u00B7"\n /** The character ½
*/\n public const val half: Char = "\u00BD"\n /** The character – */\n public const val ndash: Char =
"\u2013"\n /** The character — */\n public const val mdash:
Char = "\u2014"\n /** The character ‘ */\n public const val leftSingleQuote: Char = "\u2018"\n /**
The character ’ */\n public const val rightSingleQuote: Char = "\u2019"\n /** The character ‚
*/\n public const val lowSingleQuote: Char = "\u201A"\n /** The character “ */\n public const val
leftDoubleQuote: Char = "\u201C"\n /** The character ” */\n public const val rightDoubleQuote: Char =
"\u201D"\n /** The character „ */\n public const val lowDoubleQuote: Char = "\u201E"\n /** The

```



```

character † */\n public const val dagger: Char = "\u2020\n" /** The character ‡ */\n public
const val doubleDagger: Char = "\u2021\n" /** The character • */\n public const val bullet: Char =
"\u2022\n" /** The character … */\n public const val ellipsis: Char = "\u2026\n" /** The character
′ */\n public const val prime: Char = "\u2032\n" /** The character
″ */\n public const val doublePrime: Char = "\u2033\n" /** The character € */\n public
const val euro: Char = "\u20AC\n" /** The character ™ */\n public const val tm: Char = "\u2122\n" /**
The character ≈ */\n public const val almostEqual: Char = "\u2248\n" /** The character ≠ */\n
public const val notEqual: Char = "\u2260\n" /** The character ≤ */\n public const val lessOrEqual:
Char = "\u2264\n" /** The character ≥ */\n public const val greaterOrEqual: Char = "\u2265\n\n" /**
The character « */\n @Deprecated("This constant has a typo in the name. Use leftGuillemet instead."),
ReplaceWith("Typography.leftGuillemet("))\n @DeprecatedSinceKotlin("1.6")\n public const val
leftGuillemete: Char = "\u00AB\n\n" /** The character » */\n @Deprecated("This constant has a typo in
the name. Use rightGuillemet instead.", ReplaceWith("Typography.rightGuillemet("))\n
@DeprecatedSinceKotlin("1.6")\n public const val rightGuillemete: Char = "\u00BB\n\n" /*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n *
Represents a collection of captured groups in a single match of a regular expression.\n *\n * This collection has size
of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n *\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n *\n * An element of the
collection at the particular index can be `null` if the corresponding group in the regular expression is optional
and\n *\n * there was no match captured by that group.\n */\npublic interface MatchGroupCollection :
Collection<MatchGroup?> {\n\n /** Returns a group with the specified [index].\n *\n * @return
An instance of [MatchGroup] if the group with the specified [index] was matched or `null` otherwise.\n *\n *
Groups are indexed from 1 to the count of groups in the regular expression. A group with the index 0\n *\n
corresponds to the entire match.\n */\n public operator fun get(index: Int): MatchGroup?\n\n /**\n * Extends
[MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.\n
*/\n @SinceKotlin("1.1")\npublic interface MatchNamedGroupCollection : MatchGroupCollection {\n\n /**\n *
Returns a named group with the specified [name].\n *\n * @return An instance of [MatchGroup] if the group with the
specified [name] was matched or `null` otherwise.\n *\n * @throws IllegalArgumentException if there is no group
with the specified [name] defined in the regex pattern.\n *\n * @throws UnsupportedOperationException if getting
named groups isn't supported on the current platform.\n */\n public operator fun get(name:
String): MatchGroup?\n\n /**\n * Represents the results from a single regular expression match.\n */\npublic
interface MatchResult {\n\n /** The range of indices in the original string where match was captured.\n */\n public
val range: IntRange\n\n /** The substring from the input string captured by this match.\n */\n public val value:
String\n\n /**\n * A collection of groups matched by the regular expression.\n *\n * This collection has size
of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n *\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n */\n public val groups:
MatchGroupCollection\n\n /**\n * A list of matched indexed group values.\n *\n * This list has size of
`groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n *\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire
match.\n *\n * If the group in the regular expression is optional and there were no match captured by that
group,\n *\n * corresponding item in [groupValues] is an empty string.\n *\n * @sample
samples.text.Regexps.matchDestructuringToGroupValues\n */\n public val groupValues: List<String>\n\n /**\n *
An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment
of group values.\n *\n * component1 corresponds to the value of the first group, component2\n \u2014 of the
second, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n
public val destructured: Destructured get() = Destructured(this)\n\n /** Returns a new [MatchResult] with the
results for the next match, starting at the position\n *\n * at which the last match ended (at the character after the last

```

```

matched character).\n *\n public fun next(): MatchResult?\n /**\n * Provides components
for destructuring assignment of group values.\n *\n * [component1] corresponds to the value of the first group,
[component2] \u2014 of the second, and so on.\n *\n * If the group in the regular expression is optional and
there were no match captured by that group,\n * corresponding component value is an empty string.\n *\n *
@sample samples.text.Regexps.matchDestructuringToGroupValues\n *\n public class Destructured internal
constructor(public val match: MatchResult) {\n @kotlin.internal.InlineOnly\n public operator inline fun
component1(): String = match.groupValues[1]\n @kotlin.internal.InlineOnly\n public operator inline fun
component2(): String = match.groupValues[2]\n @kotlin.internal.InlineOnly\n public operator inline fun
component3(): String = match.groupValues[3]\n @kotlin.internal.InlineOnly\n public operator inline fun
component4(): String = match.groupValues[4]\n @kotlin.internal.InlineOnly\n
 public operator inline fun component5(): String = match.groupValues[5]\n @kotlin.internal.InlineOnly\n
 public operator inline fun component6(): String = match.groupValues[6]\n @kotlin.internal.InlineOnly\n
public operator inline fun component7(): String = match.groupValues[7]\n @kotlin.internal.InlineOnly\n
public operator inline fun component8(): String = match.groupValues[8]\n @kotlin.internal.InlineOnly\n
public operator inline fun component9(): String = match.groupValues[9]\n @kotlin.internal.InlineOnly\n
public operator inline fun component10(): String = match.groupValues[10]\n\n /**\n * Returns
destructured group values as a list of strings.\n * First value in the returned list corresponds to the value of the
first group, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n
*/\n public fun
toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)\n }\n}", "/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass()\n@file:kotlin.jvm.JvmName("DurationUnitKt")\n\npackage
kotlin.time\n\n/**\n * The list of possible time measurement units, in which a duration can be expressed.\n *\n *
The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24
[HOURS].\n *\n @SinceKotlin("1.6")\n @WasExperimental(ExperimentalTime::class)\n public expect enum class
DurationUnit {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of a microsecond.\n *\n
NANOSECONDS,\n /**\n * Time unit representing one microsecond, which is 1/1000 of a millisecond.\n
*\n MICROSECONDS,\n /**\n * Time unit representing
one millisecond, which is 1/1000 of a second.\n *\n MILLISECONDS,\n /**\n * Time unit representing
one second.\n *\n SECONDS,\n /**\n * Time unit representing one minute.\n *\n MINUTES,\n
/**\n * Time unit representing one hour.\n *\n HOURS,\n /**\n * Time unit representing one day, which
is always equal to 24 hours.\n *\n DAYS;\n}\n\n/**\n * Converts the given time duration [value] expressed in the
specified [sourceUnit] into the specified [targetUnit].\n *\n @SinceKotlin("1.3")\n internal expect fun
convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double\n\n// overflown
result is unspecified\n @SinceKotlin("1.5")\n internal expect fun convertDurationUnitOverflow(value: Long,
sourceUnit: DurationUnit, targetUnit: DurationUnit): Long\n\n// overflown result is coerced in the Long range
boundaries\n @SinceKotlin("1.5")\n internal expect fun convertDurationUnit(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit):
Long\n\n\n @SinceKotlin("1.3")\n @Suppress("REDUNDANT_ELSE_IN_WHEN")\n internal fun
DurationUnit.shortName(): String = when (this) {\n DurationUnit.NANOSECONDS -> "ns"\n
DurationUnit.MICROSECONDS -> "us"\n DurationUnit.MILLISECONDS -> "ms"\n
DurationUnit.SECONDS -> "s"\n DurationUnit.MINUTES -> "m"\n DurationUnit.HOURS -> "h"\n
DurationUnit.DAYS -> "d"\n else -> error("Unknown unit: $this")\n}\n\n @SinceKotlin("1.5")\n internal fun
durationUnitByShortName(shortName: String): DurationUnit = when (shortName) {\n "ns" ->
DurationUnit.NANOSECONDS\n "us" -> DurationUnit.MICROSECONDS\n "ms" ->
DurationUnit.MILLISECONDS\n "s" -> DurationUnit.SECONDS\n "m" -> DurationUnit.MINUTES\n
"h" -> DurationUnit.HOURS\n "d" -> DurationUnit.DAYS\n else -> throw

```

```

IllegalArgumentException("Unknown duration unit short name:
$shortName")\n}\n\n@SinceKotlin("1.5")\ninternal fun durationUnitByIsoChar(isoChar:
Char, isTimeComponent: Boolean): DurationUnit =\n when {\n !isTimeComponent -> {\n when
(isoChar) {\n 'D' -> DurationUnit.DAYS\n else -> throw IllegalArgumentException("Invalid or
unsupported duration ISO non-time unit: $isoChar")\n }\n } else -> {\n when (isoChar) {\n
 'H' -> DurationUnit.HOURS\n 'M' -> DurationUnit.MINUTES\n 'S' ->
DurationUnit.SECONDS\n else -> throw IllegalArgumentException("Invalid duration ISO time unit:
$isoChar")\n }\n }\n },"/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin.time\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n *
This annotation marks the experimental preview of
the standard library API for measuring time and working with durations.\n */\n * > Note that this API is in a preview
state and has a very high chance of being changed in the future.\n * Do not use it if you develop a library since your
library will become binary incompatible\n * with the future versions of the standard library.\n */\n * Any usage of a
declaration annotated with `@ExperimentalTime` must be accepted either by\n * annotating that usage with the
[OptIn] annotation, e.g. `@OptIn(ExperimentalTime::class)`,\n * or by using the compiler argument `-opt-
in=kotlin.time.ExperimentalTime`.\n */\n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(\n
CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n
VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n
PROPERTY_SETTER,\n TYPEALIAS)\n\n@SinceKotlin("1.3")\npublic annotation class
ExperimentalTime\n"/**\n
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage
kotlin.time\n\n/**\n * A source of time for measuring time intervals.\n */\n * The only operation provided by the
time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.\n */\n * @see
[measureTime]\n * @see [measureTimedValue]\n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic
interface TimeSource {\n /**\n * Marks a point in time on this time source.\n */\n * The returned
[TimeMark] instance encapsulates the captured time point and allows querying\n * the duration of time interval
[elapsed][TimeMark.elapsedNow] from that point.\n */\n * public fun markNow(): TimeMark\n\n /**\n *
The most precise time source available in the platform.\n */\n * This time source returns its readings from a
source of monotonic time when it is available in a target platform,\n * and resorts to a non-monotonic time source
otherwise.\n */\n * public object Monotonic : TimeSource by MonotonicTimeSource {\n override fun
toString(): String = MonotonicTimeSource.toString()\n }\n\n * public companion object {\n /**\n *
Represents a time point notched on a particular [TimeSource]. Remains bound to the time source it was taken from\n *
and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).\n */\n * @SinceKotlin("1.3")\n * @ExperimentalTime\n * public abstract class TimeMark {\n /**\n * Returns the
amount of time passed from this mark measured with the time source from which this mark was taken.\n */\n * \n * Note that the value returned by this function can change on subsequent invocations.\n */\n * public abstract fun
elapsedNow(): Duration\n /**\n * Returns a time mark on the same time source that is
ahead of this time mark by the specified [duration].\n */\n * The returned time mark is more _late_ when the
[duration] is positive, and more _early_ when the [duration] is negative.\n */\n * public open operator fun
plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)\n /**\n * Returns a time mark on
the same time source that is behind this time mark by the specified [duration].\n */\n * The returned time mark
is more _early_ when the [duration] is positive, and more _late_ when the [duration] is negative.\n */\n * public
open operator fun minus(duration: Duration): TimeMark = plus(-duration)\n /**\n * Returns true if this time
mark has passed according to the time source from which this mark was taken.\n */\n * Note that the value
returned by this function can change on subsequent invocations.\n * If the time source is monotonic, it can change

```



number of nanoseconds, \n \* thus it's capable to represent a time range of approximately \u00b11292 years. \n \*

Should the reading value overflow as the result of [plusAssign] operation, an [IllegalStateException] is thrown. \n

```

*\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic class TestTimeSource : AbstractLongTimeSource(unit
= DurationUnit.NANOSECONDS) {\n private var reading: Long = 0L\n\n override fun read(): Long =
reading\n\n /**\n * Advances the current reading value of this time source by the specified [duration]. \n *\n * [duration] value is rounded down towards zero when converting it to a [Long] number of nanoseconds. \n * For
example, if the duration being added is `0.6.nanoseconds`, the reading doesn't advance because \n * the duration
value is rounded to zero nanoseconds. \n *\n * @throws IllegalStateException when the reading value
overflows as the result of this operation. \n */\n public operator fun plusAssign(duration: Duration) {\n val
longDelta = duration.toLong(unit)\n reading = if (longDelta != Long.MIN_VALUE && longDelta !=
Long.MAX_VALUE) {\n // when delta fits in long, add it as long\n val newReading = reading +
longDelta\n if (reading xor longDelta >= 0 && reading xor newReading < 0) overflow(duration)\n
 newReading\n } else {\n val delta = duration.toDouble(unit)\n // when delta is greater than
long, add it as double\n val newReading = reading + delta\n if (newReading > Long.MAX_VALUE ||
newReading < Long.MIN_VALUE) overflow(duration)\n newReading.toLong()\n }\n }\n\n private
fun overflow(duration: Duration) {\n throw IllegalStateException("TestTimeSource will overflow if its reading
${reading}ns is advanced by $duration.")\n }\n}\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. \n */\n\npackage kotlin.time\n\nimport kotlin.contracts.*\n\n/**\n * Executes
the given function [block] and returns the duration of elapsed time interval. \n * \n * The elapsed time is measured
with [TimeSource.Monotonic]. \n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic
inline fun measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n return TimeSource.Monotonic.measureTime(block)\n}\n\n/**\n * Executes
the given function [block] and returns the duration of elapsed time interval. \n * \n * The elapsed time is
measured with the specified `this` [TimeSource] instance. \n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic
inline fun TimeSource.measureTime(block: () -> Unit):
Duration {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n val mark =
markNow()\n block()\n return mark.elapsedNow()\n}\n\n/**\n * Data class representing a result of executing
an action, along with the duration of elapsed time interval. \n * \n * @property value the result of the action. \n *
@property duration the time elapsed to execute the action. \n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic data class TimedValue<T>(val value:
T, val duration: Duration)\n\n/**\n * Executes the given function [block] and returns an instance of [TimedValue]
class, containing both \n * the result of the function execution and the duration of elapsed time interval. \n * \n * The
elapsed time is measured with [TimeSource.Monotonic]. \n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic
inline fun <T> measureTimedValue(block: () -> T): TimedValue<T> {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n\n return
TimeSource.Monotonic.measureTimedValue(block)\n}\n\n/**\n * Executes the given [block] and returns an
instance of [TimedValue] class, containing both \n * the result of function execution and the duration of elapsed time
interval. \n * \n * The elapsed time is measured with the specified `this` [TimeSource] instance. \n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic
inline fun <T> TimeSource.measureTimedValue(block: ()
-> T): TimedValue<T> {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n\n val mark = markNow()\n val result = block()\n return
TimedValue(result, mark.elapsedNow())\n}\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. \n */\n\npackage kotlin\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.native.concurrent.SharedImmutable\n\n/**\n * Defines deep recursive
function that keeps its stack on the heap, \n * which allows very deep recursive computations that do not use the
actual call stack. \n * To initiate a call to this deep recursive function use its [invoke] function. \n * As a rule of

```





```

*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Byte.floorDiv(other: Short): Int = \n
this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_
and has the absolute value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Short): Short = \n
this.toInt().mod(other.toInt()).toShort()\n\n/**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun
Byte.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Int): Int = \n
this.toInt().mod(other)\n\n/**\n * Divides this value by the other value, flooring the result to an integer that is closer to
negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic
inline fun Byte.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign
as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Long): Long = \n
this.toLong().mod(other)\n\n/**\n * Divides this value by the other value, flooring the result to an integer that is closer
to negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other:
Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute
value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Byte): Byte = \n
this.toInt().mod(other.toInt()).toByte()\n\n/**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun
Short.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Short): Short = \n
this.toInt().mod(other.toInt()).toShort()\n\n/**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline
fun Short.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Int): Int = \n
this.toInt().mod(other)\n\n/**\n * Divides this value by the other value, flooring the result to an integer that is closer
to negative infinity. *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other:
Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute
value less than the absolute value of the divisor.\n *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic
inline fun Short.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/**\n * Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity.
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Byte): Int = \n
this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n *^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun
Int.mod(other: Byte): Byte = \n this.mod(other.toInt()).toByte()\n\n/**\n * Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.

```



```

*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Short): Int = \n
this.floorDiv(other.toInt())\n\n/**\n * Calculates
the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same
sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other: Short): Short = \n
this.mod(other.toInt()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is
closer to negative infinity. *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Int.floorDiv(other: Int): Int {\n var q = this / other\n if (this xor other < 0 && q * other != this) q-- \n return
q\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is
either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the
divisor.\n *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other:
Int): Int {\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 31))\n}\n\n/** Divides
this value by the other value, flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Long): Long = \n
this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Int.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/** Divides this value by the other value, flooring
the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.floorDiv(other: Byte): Long = \n
this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or
has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.mod(other: Byte): Byte = \n
this.mod(other.toLong()).toByte()\n\n/** Divides this value by the other value, flooring the result to an integer that
is closer to negative infinity. *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.floorDiv(other: Short): Long = \n this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.mod(other: Short): Short
= \n this.mod(other.toLong()).toShort()\n\n/** Divides this value by the other value, flooring the result to an
integer that is closer to negative infinity. *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.floorDiv(other: Int): Long = \n this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.mod(other: Int): Int = \n
this.mod(other.toLong()).toInt()\n\n/** Divides this value by the other value, flooring the result to an integer that is
closer to negative infinity. *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.floorDiv(other: Long): Long {\n var q = this / other\n if (this xor other < 0 && q * other != this) q-- \n
return
q\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is
either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the
divisor.\n *\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.mod(other: Long): Long
{\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 63))\n}\n\n/**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign
as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\n

```

```

*\/n@SinceKotlin("1.5")\/n@kotlin.internal.InlineOnly\/npublic inline fun Float.mod(other: Float):
Float {\/n val r = this % other\/n return if (r != 0.0.toFloat() && r.sign != other.sign) r + other else r\/n}\/n\/n**\/n *
Calculates the remainder of flooring division of this value by the other value.\/n *\/n * The result is either zero or has
the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\/n *\/n * If the
result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value
of the result can be less than or _equal to_ the absolute value of the divisor.\/n
*\/n@SinceKotlin("1.5")\/n@kotlin.internal.InlineOnly\/npublic inline fun Float.mod(other: Double): Double = \/n
this.toDouble().mod(other)\/n\/n**\/n * Calculates the remainder of flooring division of this value by the other
value.\/n *\/n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\/n *\/n * If the result cannot be represented exactly,
it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or
equal to the absolute value of the divisor.\/n
*\/n@SinceKotlin("1.5")\/n@kotlin.internal.InlineOnly\/npublic inline
fun Double.mod(other: Float): Double = \/n this.mod(other.toDouble())\/n\/n**\/n * Calculates the remainder of
flooring division of this value by the other value.\/n *\/n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\/n *\/n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\/n
*\/n@SinceKotlin("1.5")\/n@kotlin.internal.InlineOnly\/npublic inline fun Double.mod(other: Double): Double {\/n
val r = this % other\/n return if (r != 0.0 && r.sign != other.sign) r + other else r\/n}\/n\/n",**\/n * Copyright 2010-
2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n
*\/n@SinceKotlin("1.3")\/n@kotlin.internal.InlineOnly\/npublic inline fun Any?.hashCode(): Int =
this?.hashCode() ?: 0\/n",**\/n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\/n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\/n
*\/n@SinceKotlin("1.1")\/npublic
class KotlinVersion(val major: Int, val minor: Int, val patch: Int) : Comparable<KotlinVersion> {\/n /**\/n *
Creates a version from [major] and [minor] components, leaving [patch] component zero.\/n *\/n public
constructor(major: Int, minor: Int) : this(major, minor, 0)\/n private val version = versionOf(major, minor,
patch)\/n private fun versionOf(major: Int, minor: Int, patch: Int): Int {\/n require(major in
0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE && patch in
0..MAX_COMPONENT_VALUE) {\/n "Version components are out of range: $major.$minor.$patch"\/n
}\/n return major.shl(16) + minor.shl(8) + patch\/n }\/n /**\/n * Returns the string representation of this
version\/n *\/n override fun toString(): String = "$major.$minor.$patch"\/n\/n override fun equals(other:
Any?): Boolean {\/n if (this === other) return true\/n val otherVersion = (other as? KotlinVersion) ?: return
false\/n
 return this.version == otherVersion.version\/n }\/n\/n override fun hashCode(): Int = version\/n\/n override
fun compareTo(other: KotlinVersion): Int = version - other.version\/n\/n /**\/n * Returns `true` if this version is
not less than the version specified\/n * with the provided [major] and [minor] components.\/n *\/n public fun
isAtLeast(major: Int, minor: Int): Boolean = // this.version >= versionOf(major, minor, 0)\/n this.major > major ||
(this.major == major &&\/n this.minor >= minor)\/n\/n /**\/n * Returns `true` if this version is not less
than the version specified\/n * with the provided [major], [minor] and [patch] components.\/n *\/n public fun
isAtLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor, patch)\/n
this.major > major || (this.major == major &&\/n (this.minor > minor || this.minor == minor &&\/n

```

```

 this.patch >= patch))\n\n
 companion object {\n /**\n * Maximum value a version component can have, a constant value 255.\n *\n // NOTE: Must be placed before CURRENT because its initialization requires this field being initialized
in JS\n public const val MAX_COMPONENT_VALUE = 255\n\n /**\n * Returns the current version
of the Kotlin standard library.\n *\n @kotlin.jvm.JvmField\n public val CURRENT: KotlinVersion =
KotlinVersionCurrentValue.get()\n }\n\n// this class is ignored during classpath normalization when
considering whether to recompile dependencies in Kotlin build\nprivate object KotlinVersionCurrentValue {\n
 @kotlin.jvm.JvmStatic\n fun get(): KotlinVersion = KotlinVersion(1, 6, 21) // value is written here automatically
during build\n}"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found
in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmName("LateinitKt")\n@file:Suppress("unused")\n\npackage kotlin\n\nimport
kotlin.internal.InlineOnly\nimport kotlin.internal.AccessibleLateinitPropertyLiteral\nimport
kotlin.reflect.KProperty0\n\n/**\n * Returns `true` if this lateinit property has been assigned a value, and `false`
otherwise.\n *\n * Cannot be used in an inline function, to avoid binary compatibility issues.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\ninline val @receiver:AccessibleLateinitPropertyLiteral
KProperty0<*>.isInitialized: Boolean\n get() = throw NotImplementedError("Implementation is
intrinsic")\n}"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmName("LazyKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin\n\nimport
kotlin.reflect.KProperty\n\n/**\n
 * Represents a value with lazy initialization.\n *\n * To create an instance of [Lazy] use the [lazy] function.\n
*\npublic interface Lazy<out T> {\n /**\n * Gets the lazily initialized value of the current Lazy instance.\n *\n
Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.\n *\n public
val value: T\n\n /**\n * Returns `true` if a value for this Lazy instance has been already initialized, and `false`
otherwise.\n *\n * Once this function has returned `true` it stays `true` for the rest of lifetime of this Lazy instance.\n
*\n public fun isInitialized(): Boolean\n}\n\n/**\n * Creates a new instance of the [Lazy] that is already
initialized with the specified [value].\n *\npublic fun <T> lazyOf(value: T): Lazy<T> =
InitializedLazyImpl(value)\n\n/**\n * An extension to delegate a read-only property of type [T] to an instance of
[Lazy].\n *\n * This extension allows to use instances of Lazy for property
delegation:\n *\n`val property: String by lazy { initializer }`\n *\n@kotlin.internal.InlineOnly\npublic inline operator
fun <T> Lazy<T>.getValue(thisRef: Any?, property: KProperty<*>): T = value\n\n/**\n * Specifies how a [Lazy]
instance synchronizes initialization among multiple threads.\n *\npublic enum class LazyThreadSafetyMode {\n\n
/**\n * Locks are used to ensure that only a single thread can initialize the [Lazy] instance.\n *\n
SYNCHRONIZED,\n\n /**\n * Initializer function can be called several times on concurrent access to
uninitialized [Lazy] instance value,\n * but only the first returned value will be used as the value of [Lazy]
instance.\n *\n PUBLICATION,\n\n /**\n * No locks are used to synchronize an access to the [Lazy]
instance value; if the instance is accessed from multiple threads, its behavior is undefined.\n *\n * This mode
should not be used unless the [Lazy] instance is guaranteed never to be initialized
from more than one thread.\n *\n NONE,\n}\n\n\ninternal object UNINITIALIZED_VALUE\n\n// internal to
be called from lazy in JS\ninternal class UnsafeLazyImpl<out T>(initializer: () -> T) : Lazy<T>, Serializable {\n
 private var initializer: (() -> T)? = initializer\n private var _value: Any? = UNINITIALIZED_VALUE\n\n
 override val value: T\n get() {\n if (_value === UNINITIALIZED_VALUE) {\n _value =
initializer!()\n initializer = null\n }\n @Suppress("UNCHECKED_CAST")\n return
_value as T\n }\n\n override fun isInitialized(): Boolean = _value !== UNINITIALIZED_VALUE\n\n
 override fun toString(): String = if (isInitialized()) value.toString() else "Lazy value not initialized yet."\n\n
 private fun writeReplace(): Any = InitializedLazyImpl(value)\n}\n\ninternal class InitializedLazyImpl<out
T>(override val value: T) : Lazy<T>, Serializable {\n override fun isInitialized():

```

```

Boolean = true\n\n override fun toString(): String = value.toString()\n\n}\n\n", "/*\n * Copyright 2010-2019
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("NumbersKt")\npackage kotlin\n\n/**\n *
Counts the number of set bits in the binary representation of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary
representation of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Int.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the
most significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeHighestOneBit(): Int\n\n/**\n * Returns a number having a single bit set in the position of the least
significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeLowestOneBit(): Int\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified
[bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least
significant bits on the right side.\n * Rotating the number left by a negative bit count is the same
as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating
by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) ==
number.rotateLeft(n % 32)`\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.rotateLeft(bitCount: Int): Int\n\n/**\n * Rotates the binary representation of this [Int] number right by the
specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number
as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as
rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a
multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) ==
number.rotateRight(n % 32)`\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Int.rotateRight(bitCount: Int): Int\n\n/**\n * Counts the number of set bits in the binary representation
of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Long.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero
in the binary representation of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [Long] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Long.takeHighestOneBit(): Long\n\n/**\n * Returns a number having a single bit set in the position of
the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.takeLowestOneBit(): Long\n\n/**\n * Rotates the binary representation of this [Long] number left by the
specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as

```

the least significant bits on the right side.  
Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:  
`number.rotateLeft(-n) == number.rotateRight(n)`  
Rotating by a multiple of [Long.SIZE\_BITS] (64) returns the same number, or more generally  
`number.rotateLeft(n) == number.rotateLeft(n % 64)`

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateLeft(bitCount: Int): Long\n\n Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.
* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:
number.rotateRight(-n) == number.rotateLeft(n)
* Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally
number.rotateRight(n) == number.rotateRight(n % 64)
```

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateRight(bitCount: Int): Long\n\n Counts the number of set bits in the binary representation of this [Byte] number.
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n\n Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)\n\n Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits()\n\n Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number, or zero, if this number is zero.
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()\n\n Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number, or zero, if this number is zero.
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()\n\n Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits.
* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.
* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:
number.rotateLeft(-n) == number.rotateRight(n)
* Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally
number.rotateLeft(n) == number.rotateLeft(n % 8)
```

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateLeft(bitCount: Int): Byte =\n (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()\n\n Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits.
* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:
number.rotateRight(-n) == number.rotateLeft(n)
* Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally
number.rotateRight(n) == number.rotateRight(n % 8)
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateRight(bitCount: Int): Byte =\n (toInt()).shl(8
```

- (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n\n/\*\*\n \* Counts the number of set bits in the binary representation of this [Short] number.\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.\n
```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countLeadingZeroBits(): Int =\n (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE_BITS - Short.SIZE_BITS)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number.\n
```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countTrailingZeroBits():\n Int = (toInt() or 0x10000).countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Short] number,\n * or zero, if this number is zero.\n
```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Short] number,\n * or zero, if this number is zero.\n
```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeLowestOneBit(): Short = toInt().takeLowestOneBit().toShort()\n\n/**\n * Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
```

```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateLeft(bitCount: Int): Short =\n (toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(16 - (bitCount and 15))).toShort()\n\n/**\n * Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
```

```

\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateRight(bitCount: Int): Short =\n (toInt().shl(16 - (bitCount and 15)) or (toInt() and 0xFFFF).ushr(bitCount and 15)).toShort()\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\nimport kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.2")\n@Suppress("INVISIBLE_MEMBER", "INVISIBLE_REFERENCE")\n@RequireKotlin("1.2.30", level = DeprecationLevel.HIDDEN, versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic inline fun <R> suspend(noinline block: suspend () -> R): suspend () -> R = block\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("TuplesKt")\npackage kotlin\n\n/**\n * Represents a generic pair of two values.\n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Pair exhibits value semantics, i.e. two pairs are equal if both components are equal.\n * An example of decomposing it into values:\n * @sample samples.misc.Tuples.pairDestructuring\n
```

```

*\n * @param A type of the first value.\n * @param B type of the second value.\n * @property first First value.\n *
@property second Second value.\n * @constructor Creates a new instance of Pair.\n */\npublic data class Pair<out
A, out B>(\n public val first: A,\n public val second: B\n) : Serializable {\n /**\n * Returns
string representation of the [Pair] including its [first] and [second] values.\n */\n public override fun toString():
String = \"($first, $second)\"\n}\n\n/**\n * Creates a tuple of type [Pair] from this and [that].\n */\n * This can be
useful for creating [Map] literals with less noise, for example:\n * @sample
samples.collections.maps.instantiation.mapFromPairs\n */\npublic infix fun <A, B> A.to(that: B): Pair<A, B> =
Pair(this, that)\n\n/**\n * Converts this pair into a list.\n */\n * @sample samples.misc.tuples.pairToList\n */\npublic
fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)\n\n/**\n * Represents a triad of values\n */\n * There is
no meaning attached to values in this class, it can be used for any purpose.\n * Triple exhibits value semantics, i.e.
two triples are equal if all three components are equal.\n * An example of decomposing it into values:\n * @sample
samples.misc.tuples.tripleDestructuring\n */\n * @param A type of the first value.\n * @param B type
of the second value.\n * @param C type of the third value.\n * @property first First value.\n * @property second
Second value.\n * @property third Third value.\n */\npublic data class Triple<out A, out B, out C>(\n public val
first: A,\n public val second: B,\n public val third: C\n) : Serializable {\n /**\n * Returns string
representation of the [Triple] including its [first], [second] and [third] values.\n */\n public override fun
toString(): String = \"($first, $second, $third)\"\n}\n\n/**\n * Converts this triple into a list.\n */\n * @sample
samples.misc.tuples.tripleToList\n */\npublic fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first, second,
third)\n", "*/\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n//
Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport kotlin.internal.*\n\n/**\n * A range of values of type `UInt`.\n
*/\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic class UIntRange(start:
UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt> {\n override val start:
UInt get() = first\n override val endInclusive: UInt get() = last\n\n override fun contains(value: UInt): Boolean =
first <= value && value <= last\n\n /**\n * Checks if the range is empty.\n */\n * The range is empty if its
start value is greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun
equals(other: Any?): Boolean =\n other is UIntRange && (isEmpty() && other.isEmpty()) ||\n first ==
other.first && last == other.last\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * first.toInt()
+ last.toInt())\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n
 /**\n * An empty range of values of type UInt. */\n public val EMPTY: UIntRange =
UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)\n }\n}\n\n/**\n * A progression of values of type `UInt`.\n
*/\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class
UIntProgression\n\ninternal constructor(\n start: UInt,\n endInclusive: UInt,\n step: Int\n) : Iterable<UInt> {\n
 init {\n if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step
== Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to
avoid overflow on negation.")\n }\n\n /**\n * The first element in the progression.\n */\n public val first:
UInt = start\n\n /**\n * The last element in the progression.\n */\n public val last: UInt =
getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public
val step:
Int = step\n\n final override fun iterator(): Iterator<UInt> = UIntProgressionIterator(first, last, step)\n\n /**\n
 * Checks if the progression is empty.\n */\n * Progression with a positive step is empty if its first element is
greater than the last element.\n * Progression with a negative step is empty if its first element is less than the last
element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun
equals(other: Any?): Boolean =\n other is UIntProgression && (isEmpty() && other.isEmpty()) ||\n first ==
other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n if
(isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt()) + step.toInt())\n\n override fun toString(): String = if (step
> 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n companion object {\n
 /**\n *

```

Creates UIntProgression

within the specified bounds of a closed range.  
\* The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].  
\* In order to go backwards the [step] must be negative.  
\* [step] must be greater than `Int.MIN\_VALUE` and not equal to zero.

```
public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression =
 UIntProgression(rangeStart, rangeEnd, step)
}

/** An iterator over a progression of values of type
`UInt`.
* @property step the number by which the value is incremented on each step.
*/
@SinceKotlin("1.3")
@Suppress("DEPRECATION_ERROR")
private class UIntProgressionIterator(first:
UInt, last: UInt, step: Int) : UIntIterator() {
 private val finalElement = last
 private var hasNext: Boolean = if
(step > 0) first <= last else first >= last
 private val step = step.toUInt() // use
2-complement math for negative steps
 private var next = if (hasNext) first else finalElement
 override fun
hasNext(): Boolean = hasNext
 override fun nextUInt(): UInt {
 val value = next
 if (value ==
finalElement) {
 if (!hasNext) throw kotlin.NoSuchElementException()
 hasNext = false
 }
 else {
 next += step
 }
 return value
 }
}

/** Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.
*/
@Auto-generated file. DO NOT EDIT!
package
kotlin.collections
/** An iterator over a sequence of values of type `UByte`.
*/
@Deprecated("This class is not
going to be stabilized and is to be removed soon.", level =
DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public abstract class UByteIterator : Iterator<UByte> {
 final override
fun next() = nextUByte()
 /** Returns the next value in the sequence without boxing.
*/
 public abstract fun
nextUByte(): UByte
}

/** An iterator over a sequence of values of type `UShort`.
*/
@Deprecated("This class
is not going to be stabilized and is to be removed soon.", level =
DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public abstract class UShortIterator : Iterator<UShort> {
 final override fun
next() = nextUShort()
 /** Returns the next value in the sequence without boxing.
*/
 public abstract fun
nextUShort(): UShort
}

/** An iterator over a sequence of values of type `UInt`.
*/
@Deprecated("This class is not going to be stabilized and is to be removed soon.", level =
DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public abstract class UIntIterator : Iterator<UInt> {
 final
override fun next() = nextUInt()
 /** Returns the next value in the sequence without boxing.
*/
 public
abstract fun nextUInt(): UInt
}

/** An
iterator over a sequence of values of type `ULong`.
*/
@Deprecated("This class is not going to be stabilized and
is to be removed soon.", level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public abstract class
ULongIterator : Iterator<ULong> {
 final override fun next() = nextULong()
 /** Returns the next value in
the sequence without boxing.
*/
 public abstract fun
nextULong(): ULong
}

/** Copyright 2010-2022
JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/
@Auto-generated file. DO NOT
EDIT!
package kotlin.ranges
import kotlin.internal.*
/** A range of values of type `ULong`.
*/
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public class
ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1),
 ClosedRange<ULong> {
 override val start:
ULong get() = first
 override val endInclusive: ULong get() = last
 override fun contains(value: ULong):
Boolean = first <= value && value <= last
 /**
* Checks if the range is empty.
* The range is
empty if its start value is greater than the end value.
*/
 override fun isEmpty(): Boolean = first > last
 override fun equals(other: Any?): Boolean =
 other is ULongRange && (isEmpty() && other.isEmpty() ||
 first == other.first && last == other.last)
 override fun hashCode(): Int =
 if (isEmpty()) -1 else (31
* (first xor (first shr 32)).toInt() + (last xor (last shr 32)).toInt())
 override fun toString(): String =
"$first..$last"
 companion object {
 /** An empty range of values of type ULong.
*/
 public val
EMPTY: ULongRange = ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)
}
}

/** A
```





```

*\n\n@file:kotlin.jvm.JvmName("\UNumbersKt")\npackage kotlin\n\n/**\n * Counts the number of set bits in the
binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countOneBits():
Int = toInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countLeadingZeroBits(): Int =
toInt().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the
binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countTrailingZeroBits(): Int =
toInt().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeHighestOneBit(): UInt =
toInt().takeHighestOneBit().toUInt()\n\n/**\n * Returns a number having a single bit set in the position of the least
significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeLowestOneBit(): UInt =
toInt().takeLowestOneBit().toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number left by the
specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as
the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as
rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a
multiple of [UInt.SIZE_BITS]
(32) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateLeft(bitCount: Int):
UInt = toInt().rotateLeft(bitCount).toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number
right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter
the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is
the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *
Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()\n\n/**\n * Counts the
number of set bits in the binary representation of this [ULong] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countOneBits(): Int =
toLong().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [ULong] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countLeadingZeroBits(): Int =
toLong().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [ULong] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,

```

```

ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countTrailingZeroBits(): Int
= toLong().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeHighestOneBit(): ULong
= toLong().takeHighestOneBit().toULong()\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeLowestOneBit(): ULong
= toLong().takeLowestOneBit().toULong()\n\n/**\n * Rotates the binary representation
of this [ULong] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out
from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left
by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n *\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or
more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateLeft(bitCount:
Int): ULong = toLong().rotateLeft(bitCount).toULong()\n\n/**\n * Rotates the binary representation of this [ULong]
number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side
reenter the number as the most significant bits
on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the
negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of
[ULong.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) ==
number.rotateRight(n % 64)`\n *\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateRight(bitCount:
Int): ULong = toLong().rotateRight(bitCount).toULong()\n\n/**\n * Counts the number of set bits in the binary
representation of this [UByte] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countOneBits(): Int =
toUInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of
this [UByte] number.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countLeadingZeroBits(): Int =
toByte().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [UByte] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countTrailingZeroBits(): Int =
toByte().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeHighestOneBit(): UByte
= toInt().takeHighestOneBit().toUByte()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeLowestOneBit(): UByte =
toInt().takeLowestOneBit().toUByte()\n\n/**\n * Rotates the binary representation of this [UByte] number left by
the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the

```

number as the least significant bits on the right side.  
`number.rotateLeft(-n) == number.rotateRight(n)`  
Rotating by a multiple of [UByte.SIZE\_BITS] (8) returns the same number, or more generally  
`number.rotateLeft(n) == number.rotateLeft(n % 8)`

```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateLeft(bitCount:
Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n\n*\n * Rotates the binary representation of this [UByte]
number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side
reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit
count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 8)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateRight(bitCount:
Int): UByte = toByte().rotateRight(bitCount).toUByte()\n\n*\n
* Counts the number of set bits in the binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countOneBits(): Int =
toUInt().countOneBits()\n\n*\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countLeadingZeroBits(): Int =
toShort().countLeadingZeroBits()\n\n*\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countTrailingZeroBits():
Int = toShort().countTrailingZeroBits()\n\n*\n * Returns a number having a single bit set in the position of the
most significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeHighestOneBit(): UShort =
toInt().takeHighestOneBit().toUShort()\n\n*\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeLowestOneBit(): UShort =
toInt().takeLowestOneBit().toUShort()\n\n*\n * Rotates the binary representation of this [UShort] number left
by the specified [bitCount] number of bits.\n * The most significant bits pushed out
from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left
by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n *\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or
more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateLeft(bitCount:
Int): UShort = toShort().rotateLeft(bitCount).toUShort()\n\n*\n * Rotates the binary representation of this
[UShort] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the
right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a
negative bit count is the same as rotating it left by the
negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of
[UShort.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateRight(n) ==

```

```

number.rotateRight(n % 16)`n *^n@SinceKotlin("1.6")`n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)`n@kotlin.internal.InlineOnly`npublic inline fun UShort.rotateRight(bitCount:
Int): UShort = toShort().rotateRight(bitCount).toUShort()`n", "/*`n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.`n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.`n *^n`npackage kotlin.internal`n`n// (a - b) mod c`nprivate fun
differenceModulo(a: UInt, b: UInt, c: UInt): UInt {`n val ac = a % c`n val bc = b % c`n return if (ac >= bc) ac -
bc else ac - bc + c`n}`n`nprivate fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {`n val ac = a %
c`n val bc
= b % c`n return if (ac >= bc) ac - bc else ac - bc + c`n}`n`n/**`n * Calculates the final element of a bounded
arithmetic progression, i.e. the last element of the progression which is in the range`n * from [start] to [end] in case
of a positive [step], or from [end] to [start] in case of a negative`n * [step].`n *`n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:`n *`n * - either `step > 0` and `start <=
end`,`n * - or `step < 0` and `start >= end`.`n *`n * @param start first element of the progression`n * @param end
ending bound for the progression`n * @param step increment, or difference of successive elements in the
progression`n * @return the final element of the progression`n * @suppress`n
*^n@PublishedApi`n@SinceKotlin("1.3")`ninternal fun getProgressionLastElement(start: UInt, end: UInt, step:
Int): UInt = when {`n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())`n
step <
0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())`n else -> throw
kotlin.IllegalArgumentException("Step is zero.")`n}`n`n/**`n * Calculates the final element of a bounded
arithmetic progression, i.e. the last element of the progression which is in the range`n * from [start] to [end] in case
of a positive [step], or from [end] to [start] in case of a negative`n * [step].`n *`n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:`n *`n * - either `step > 0` and `start <=
end`,`n * - or `step < 0` and `start >= end`.`n *`n * @param start first element of the progression`n * @param end
ending bound for the progression`n * @param step increment, or difference of successive elements in the
progression`n * @return the final element of the progression`n * @suppress`n
*^n@PublishedApi`n@SinceKotlin("1.3")`ninternal fun getProgressionLastElement(start: ULong, end: ULong,
step: Long): ULong = when {`n
step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toULong())`n step < 0 -> if (start <=
end) end else end + differenceModulo(start, end, (-step).toULong())`n else -> throw
kotlin.IllegalArgumentException("Step is zero.")`n}`n`n", "/*`n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.`n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.`n *^n`n@file:kotlin.jvm.JvmName("UStringsKt") // string representation
of unsigned numbers`n`npackage kotlin.text`n`n/**`n * Returns a string representation of this [Byte] value in the
specified [radix].`n *`n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string
conversion.`n
*^n@SinceKotlin("1.5")`n@WasExperimental(ExperimentalUnsignedTypes::class)`n`n@kotlin.internal.InlineOnly
`npublic /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)`n`n`n/**`n * Returns
a string representation of this [Short] value in the specified [radix].`n *`n * @throws IllegalArgumentException
when [radix] is not a valid radix for number to string conversion.`n
*^n@SinceKotlin("1.5")`n@WasExperimental(ExperimentalUnsignedTypes::class)`n`n@kotlin.internal.InlineOnly
`npublic /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)`n`n`n/**`n * Returns a string
representation of this [Int] value in the specified [radix].`n *`n * @throws IllegalArgumentException when [radix] is
not a valid radix for number to string conversion.`n
*^n@SinceKotlin("1.5")`n@WasExperimental(ExperimentalUnsignedTypes::class)`n`n@kotlin.internal.InlineOnly
`npublic /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)`n`n`n/**`n * Returns a string
representation of this [Long] value in the specified [radix].`n *`n * @throws IllegalArgumentException when [radix]
is not a valid radix for number to string conversion.`n

```

```

*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic
fun ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))^n^n/**
 * Parses the string as a signed [UByte] number and returns the result.
 * @throws NumberFormatException if the string is not a
valid representation of a number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun String.toUByte():
UByte = toUByteOrNull() ?: numberFormatException(this)^n^n/**
 * Parses the string as a signed [UByte] number and
returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatException(this)^n^n/**
 * Parses the
string as a [UShort] number and
returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun String.toUShort():
UShort = toUShortOrNull() ?: numberFormatException(this)^n^n/**
 * Parses the string as a [UShort] number and
returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUShort(radix: Int): UShort = toUShortOrNull(radix) ?: numberFormatException(this)^n^n/**
 * Parses the
string as an [UInt] number and returns the result.
 * @throws NumberFormatException if the string is not a valid
representation of a number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun String.toUInt():
UInt
= toUIntOrNull() ?: numberFormatException(this)^n^n/**
 * Parses the string as an [UInt] number and returns the
result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUInt(radix: Int): UInt = toUIntOrNull(radix) ?: numberFormatException(this)^n^n/**
 * Parses the string as a
[ULong] number and returns the result.
 * @throws NumberFormatException if the string is not a valid
representation of a number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun String.toULong():
ULong = toULongOrNull() ?: numberFormatException(this)^n^n/**
 * Parses the string as a [ULong] number and
returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws
IllegalArgumentException
when [radix] is not a valid radix for string to number conversion.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toULong(radix: Int): ULong = toULongOrNull(radix) ?: numberFormatException(this)^n^n^n^n/**
 * Parses
the string as an [UByte] number and returns the result
 * or `null` if the string is not a valid representation of a
number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUByteOrNull(): UByte? = toUByteOrNull(radix = 10)^n^n/**
 * Parses the string as an [UByte] number
and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUByteOrNull(radix: Int): UByte? {
 val int = this.toUIntOrNull(radix) ?: return null
 if (int > UByte.MAX_VALUE) return null
 return int.toUByte()
}^n^n/**
 * Parses the string as an [UShort]
number and returns the result
 * or `null` if the string is not a valid representation of a number.
^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun
String.toUShortOrNull(): UShort? = toUShortOrNull(radix = 10)^n^n/**
 * Parses the string as an [UShort] number
and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws

```

```

IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShortOrNull(radix: Int): UShort? {\n val int = this.toUIntOrNull(radix) ?: return null\n if (int >
UShort.MAX_VALUE) return null\n return int.toUShort()\n}\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not
a valid representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(): UInt? = toUIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(radix: Int): UInt? {\n checkRadix(radix)\n\n val length = this.length\n if (length == 0)
return null\n\n val limit: UInt = UInt.MAX_VALUE\n val start: Int\n val firstChar = this[0]\n if (firstChar
< '0') {\n if (length == 1 || firstChar != '+') return null\n start = 1\n } else {\n start = 0\n }\n\n val
limitForMaxRadix = 119304647u // limit / 36\n var limitBeforeMul = limitForMaxRadix\n
 val uradix = radix.toUInt()\n var result = 0u\n for (i in start until length) {\n val digit = digitOf(this[i],
radix)\n\n if (digit < 0) return null\n if (result > limitBeforeMul) {\n if (limitBeforeMul ==
limitForMaxRadix) {\n limitBeforeMul = limit / uradix\n\n if (result > limitBeforeMul) {\n
 return null\n }\n } else {\n return null\n }\n }\n result *= uradix\n
 val beforeAdding = result\n result += digit.toUInt()\n if (result < beforeAdding) return null // overflow
has happened\n }\n\n return result\n}\n\n/**\n * Parses the string as an [ULong] number and returns the result\n
* or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)\n\n/**\n * Parses the string as an [ULong] number and returns the result\n
* or `null` if the string is not a valid representation
of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number
conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(radix: Int): ULong? {\n checkRadix(radix)\n\n val length = this.length\n if (length ==
0) return null\n\n val limit: ULong = ULong.MAX_VALUE\n val start: Int\n val firstChar = this[0]\n if
(firstChar < '0') {\n if (length == 1 || firstChar != '+') return null\n start = 1\n } else {\n start = 0\n }\n\n val
limitForMaxRadix = 512409557603043100uL // limit / 36\n var limitBeforeMul =
limitForMaxRadix\n val uradix = radix.toULong()\n var result = 0uL\n for (i in start until length) {\n val
digit = digitOf(this[i], radix)\n\n if (digit < 0) return null\n
 if (result > limitBeforeMul) {\n if (limitBeforeMul == limitForMaxRadix) {\n limitBeforeMul =
limit / uradix\n\n if (result > limitBeforeMul) {\n return null\n }\n }
else {\n return null\n }\n }\n result *= uradix\n val beforeAdding = result\n
 result += digit.toUInt()\n if (result < beforeAdding) return null // overflow has happened\n }\n\n return
result\n}\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@file:Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\npackage kotlin\n\nimport
kotlin.annotation.AnnotationTarget.\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\n\n/**\n * Marks the API that is dependent on the
experimental unsigned types, including those types themselves.\n * @Usages of such API will be reported as
warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g.
`@OptIn(ExperimentalUnsignedTypes::class)`\n * or with the `--opt-in=kotlin.ExperimentalUnsignedTypes`
compiler option is given.\n * It's recommended to propagate the experimental status to the API that depends on
unsigned types by annotating it with this annotation.\n *\n@RequiresOptIn(level =
RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Target(CLASS, ANNOTATION_CLASS,

```

PROPERTY, FIELD, LOCAL\_VARIABLE, VALUE\_PARAMETER, CONSTRUCTOR, FUNCTION,  
PROPERTY\_GETTER, PROPERTY\_SETTER,

TYPEALIAS)\n@Retention(AnnotationRetention.BINARY)\n@RequireKotlin("1.2.50", versionKind =  
RequireKotlinVersionKind.COMPILER\_VERSION)\npublic annotation class ExperimentalUnsignedTypes\n", /\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code  
is governed by

the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

\*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MathKt")\n\n\npackage

kotlin.math\n\n\n// constants, can't use them from nativeMath as they are not constants there\n\n/\*\* Ratio of the  
circumference of a circle to its diameter, approximately 3.14159. \*\n\n@SinceKotlin("1.2")\n\npublic const val PI:  
Double = 3.141592653589793\n\n/\*\* Base of the natural logarithms, approximately 2.71828.

\*\n\n@SinceKotlin("1.2")\n\npublic const val E: Double = 2.718281828459045\n\n\n// region =====  
Double Math =====\n\n\n/\*\* Computes the sine of the angle [x]  
given in radians.\n \* \n \* Special cases:\n \* - `sin(NaN|+Inf|-Inf)` is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun sin(x: Double): Double\n\n\n/\*\* Computes the cosine of the angle [x] given in radians.\n \* \n \* Special  
cases:\n \* - `cos(NaN|+Inf|-Inf)` is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect

fun cos(x: Double): Double\n\n\n/\*\* Computes the tangent of the angle [x] given in radians.\n \* \n \* Special cases:\n \* - `tan(NaN|+Inf|-Inf)` is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun tan(x: Double): Double\n\n\n/\*\* Computes the arc sine of the value [x];\n \* the returned value is an angle in the range from  $-\pi/2$  to  $\pi/2$   
radians.\n \* \n \* Special cases:\n \* - `asin(x)` is `NaN`, when  $abs(x) > 1$  or x is `NaN`\n

\*\n\n@SinceKotlin("1.2")\n\npublic expect fun asin(x: Double): Double\n\n\n/\*\* Computes the arc cosine of the  
value [x];\n \* the returned value is an angle in the range from  $0.0$  to  $\pi$  radians.\n \* \n \* Special cases:\n \* -  
`acos(x)` is `NaN`, when  $abs(x) > 1$  or x is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun acos(x: Double):

Double\n\n\n/\*\* Computes the arc tangent of the value [x];\n \* the returned value is an angle in the range from  $-\pi/2$   
to  $\pi/2$  radians.\n \* \n \* Special cases:\n \* - `atan(NaN)` is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun atan(x: Double): Double\n\n\n/\*\* Returns the angle `theta` of the polar coordinates `(r, theta)` that  
correspond\n \* to the rectangular coordinates `(x, y)` by computing the arc tangent of the value  $y / x$ ;\n \* the  
returned value is an angle in the range from  $-\pi$  to  $\pi$  radians.\n \* \n \* Special cases:\n \* - `atan2(0.0, 0.0)` is  
 $0.0$ \n \* - `atan2(0.0, x)` is  $0.0$  for  $x > 0$  and  $\pi$  for  $x < 0$ \n \* - `atan2(-0.0, x)` is  $-0.0$  for  $x > 0$  and  $-\pi$   
for  $x < 0$ \n \* - `atan2(y, +Inf)` is  $0.0$  for  $0 < y < +Inf$  and  $-0.0$  for  $-Inf < y < 0$ \n \* - `atan2(y, -Inf)` is  $\pi$   
for  $0 < y < +Inf$  and  $-\pi$  for  $-Inf < y < 0$ \n \* - `atan2(y, 0.0)` is  $\pi/2$  for  $y > 0$  and  $-\pi/2$  for  $y < 0$ \n \* -  
`atan2(+Inf, x)` is  $\pi/2$  for finite  $x$ \n \* - `atan2(-Inf, x)` is  $-\pi/2$  for finite  $x$ \n \* - `atan2(NaN, x)` and  
`atan2(y, NaN)` is `NaN`\n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun atan2(y: Double, x: Double):

Double\n\n\n/\*\* Computes  
the hyperbolic sine of the value [x].\n \* \n \* Special cases:\n \* - `sinh(NaN)` is `NaN`\n \* - `sinh(+Inf)` is  $+Inf$ \n \* -  
`sinh(-Inf)` is  $-Inf$ \n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun sinh(x: Double): Double\n\n\n/\*\* Computes the hyperbolic cosine of the value [x].\n \* \n \* Special cases:\n \* - `cosh(NaN)` is `NaN`\n \* -  
`cosh(+Inf|-Inf)` is  $+Inf$ \n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun cosh(x: Double): Double\n\n\n/\*\* Computes the hyperbolic tangent of the value [x].\n \* \n \* Special cases:\n \* - `tanh(NaN)` is `NaN`\n \* -  
`tanh(+Inf)` is  $1.0$ \n \* - `tanh(-Inf)` is  $-1.0$ \n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun tanh(x: Double):

Double\n\n\n/\*\* Computes the inverse hyperbolic sine of the value [x].\n \* \n \* The returned value is `y` such that  
 $sinh(y) == x$ .\n \* \n \* Special cases:\n \* - `asinh(NaN)` is `NaN`\n \* - `asinh(+Inf)` is  $+Inf$ \n \* - `asinh(-Inf)`  
is  $-Inf$ \n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun asinh(x: Double): Double\n\n\n/\*\* Computes the inverse hyperbolic cosine of the value [x].\n \* \n \* The returned value is positive `y` such that  
 $cosh(y) == x$ .\n \* \n \* Special cases:\n \* - `acosh(NaN)` is `NaN`\n \* - `acosh(x)` is `NaN` when  $x < 1$ \n \* -  
`acosh(+Inf)` is  $+Inf$ \n \* \n\n@SinceKotlin("1.2")\n\npublic expect fun acosh(x: Double): Double\n\n\n/\*\* Computes the inverse hyperbolic tangent of the value [x].\n \* \n \* The returned value is `y` such that  $tanh(y) ==$   
 $x$ .\n \* \n \* Special cases:\n \* - `tanh(NaN)` is `NaN`\n \* - `tanh(x)` is `NaN` when  $x > 1$  or  $x < -1$ \n \* -



`tanh(1.0)` is `+Inf` \* `-tanh(-1.0)` is `-Inf` \* `@SinceKotlin("1.2")` `public expect fun atanh(x: Double): Double` \* Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow. \* Special cases: \* `-` returns `+Inf` if any of arguments is infinite \* `-` returns `NaN` if any of arguments is `NaN` and the other is not infinite \* `@SinceKotlin("1.2")` `public expect fun hypot(x: Double, y: Double): Double` \* Computes the positive square root of the value `[x]`. \* Special cases: \* `-` `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN` \* `@SinceKotlin("1.2")` `public expect fun sqrt(x: Double): Double` \* Computes Euler's number `e` raised to the power of the value `[x]`. \* Special cases: \* `-` `exp(NaN)` is `NaN` \* `-` `exp(+Inf)` is `+Inf` \* `-` `exp(-Inf)` is `0.0` \* `@SinceKotlin("1.2")` `public expect fun exp(x: Double): Double` \* Computes `exp(x) - 1`. \* This function can be implemented to produce more precise result for `[x]` near zero. \* Special cases: \* `-` `expm1(NaN)` is `NaN` \* `-` `expm1(+Inf)` is `+Inf` \* `-` `expm1(-Inf)` is `-1.0` \* `@see [exp] function.` \* `@SinceKotlin("1.2")` `public expect fun expm1(x: Double): Double` \* Computes the logarithm of the value `[x]` to the given `[base]`. \* Special cases: \* `-` `log(x, b)` is `NaN` if either `x` or `b` are `NaN` \* `-` `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0` \* `-` `log(+Inf, +Inf)` is `NaN` \* `-` `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1` \* `-` `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1` \* `@see also` logarithm functions for common fixed bases: `[ln]`, `[log10]` and `[log2]`. \* `@SinceKotlin("1.2")` `public expect fun log(x: Double, base: Double): Double` \* Computes the natural logarithm (base `E`) of the value `[x]`. \* Special cases: \* `-` `ln(NaN)` is `NaN` \* `-` `ln(x)` is `NaN` when `x < 0.0` \* `-` `ln(+Inf)` is `+Inf` \* `-` `ln(0.0)` is `-Inf` \* `@SinceKotlin("1.2")` `public expect fun ln(x: Double): Double` \* Computes the common logarithm (base 10) of the value `[x]`. \* `@see [ln] function for special cases.` \* `@SinceKotlin("1.2")` `public expect fun log10(x: Double): Double` \* Computes the binary logarithm (base 2) of the value `[x]`. \* `@see [ln] function for special cases.` \* `@SinceKotlin("1.2")` `public expect fun log2(x: Double): Double` \* Computes `ln(x + 1)`. \* This function can be implemented to produce more precise result for `[x]` near zero. \* Special cases: \* `-` `ln1p(NaN)` is `NaN` \* `-` `ln1p(x)` is `NaN` where `x < -1.0` \* `-` `ln1p(-1.0)` is `-Inf` \* `-` `ln1p(+Inf)` is `+Inf` \* `@see [ln] function` \* `@see [expm1] function` \* `@SinceKotlin("1.2")` `public expect fun ln1p(x: Double): Double` \* Rounds the given value `[x]` to an integer towards positive infinity. \* `@return` the smallest double value that is greater than or equal to the given value `[x]` and is a mathematical integer. \* Special cases: \* `-` `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \* `@SinceKotlin("1.2")` `public expect fun ceil(x: Double): Double` \* Rounds the given value `[x]` to an integer towards negative infinity. \* `@return` the largest double value that is smaller than or equal to the given value `[x]` and is a mathematical integer. \* Special cases: \* `-` `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \* `@SinceKotlin("1.2")` `public expect fun floor(x: Double): Double` \* Rounds the given value `[x]` to an integer towards zero. \* `@return` the value `[x]` having its fractional part truncated. \* Special cases: \* `-` `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \* `@SinceKotlin("1.2")` `public expect fun truncate(x: Double): Double` \* Rounds the given value `[x]` towards the closest integer with ties rounded towards even integer. \* Special cases: \* `-` `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \* `@SinceKotlin("1.2")` `public expect fun round(x: Double): Double` \* Returns the absolute value of the given value `[x]`. \* Special cases: \* `-` `abs(NaN)` is `NaN` \* `@see absoluteValue extension property for [Double]` \* `@SinceKotlin("1.2")` `public expect fun abs(x: Double): Double` \* Returns the sign of the given value `[x]`: \* `-` `-1.0` if the value is negative, \* `-` zero if the value is zero, \* `-` `1.0` if the value is positive \* Special case: \* `-` `sign(NaN)` is `NaN` \* `@SinceKotlin("1.2")` `public expect fun sign(x: Double): Double` \* Returns the smaller of two values. \* If either value is `NaN`, then the result is `NaN`. \* `@SinceKotlin("1.2")` `public expect fun min(a: Double, b: Double): Double` \* Returns the greater of two values. \* If either value is `NaN`, then the result is `NaN`. \* `@SinceKotlin("1.2")` `public expect fun max(a: Double, b: Double): Double` // extensions \* Raises this value to the power `[x]`. \* Special cases: \* `-` `b.pow(0.0)` is `1.0` \* `-`

`b.pow(1.0) == b`n * - `b.pow(NaN)` is `NaN`n * - `NaN.pow(x)` is  
`NaN` for `x != 0.0`n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`n * - `b.pow(x)` is `NaN` for `b < 0` and `x`  
is finite and not an integern */n@SinceKotlin("1.2")npublic expect fun Double.pow(x: Double): Doublen/n/**n  
* Raises this value to the integer power [n].n */n * See the other overload of [pow] for details.n  
*/n@SinceKotlin("1.2")npublic expect fun Double.pow(n: Int): Doublen/n/**n * Returns the absolute value of  
this value.n */n * Special cases:n * - `NaN.absoluteValue` is `NaN`n */n * @see abs functionn  
*/n@SinceKotlin("1.2")npublic expect val Double.absoluteValue: Doublen/n/**n * Returns the sign of this  
value:n * - `-1.0` if the value is negative,n * - zero if the value is zero,n * - `1.0` if the value is positiven */n *  
Special case:n * - `NaN.sign` is `NaN`n */n@SinceKotlin("1.2")npublic expect val Double.sign:  
Doublen/n/**n * Returns this value with the sign bit same as of the [sign] value.n */n * If [sign] is `NaN`  
the sign of the result is undefined.n */n@SinceKotlin("1.2")npublic expect fun Double.withSign(sign: Double):  
Doublen/n/**n * Returns this value with the sign bit same as of the [sign] value.n  
*/n@SinceKotlin("1.2")npublic expect fun Double.withSign(sign: Int): Doublen/n/**n * Returns the ulp (unit in  
the last place) of this value.n */n * An ulp is a positive distance between this value and the next nearest [Double]  
value larger in magnitude.n */n * Special Cases:n * - `NaN.ulp` is `NaN`n * - `x.ulp` is `+Inf` when `x` is `+Inf`  
or `-Inf`n * - `0.0.ulp` is `Double.MIN_VALUE`n */n@SinceKotlin("1.2")npublic expect val Double.ulp:  
Doublen/n/**n * Returns the [Double] value nearest to this value in direction of positive infinity.n  
*/n@SinceKotlin("1.2")npublic expect fun Double.nextUp(): Doublen/n/**n * Returns the [Double] value  
nearest to this value in direction of negative infinity.n */n@SinceKotlin("1.2")npublic expect fun  
Double.nextDown():  
Doublen/n/**n * Returns the [Double] value nearest to this value in direction from this value towards the value  
[to].n */n * Special cases:n * - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`n * -  
`x.nextTowards(x) == x`n */n */n@SinceKotlin("1.2")npublic expect fun Double.nextTowards(to: Double):  
Doublen/n/**n * Rounds this [Double] value to the nearest integer and converts the result to [Int].n * Ties are  
rounded towards positive infinity.n */n * Special cases:n * - `x.roundToInt() == Int.MAX_VALUE` when `x >  
Int.MAX_VALUE`n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`n */n * @throws  
IllegalArgumentException when this value is `NaN`n */n@SinceKotlin("1.2")npublic expect fun  
Double.roundToInt(): Intn/n/**n * Rounds this [Double] value to the nearest integer and converts the result to  
[Long].n * Ties are rounded towards positive infinity.n */n * Special cases:n * - `x.roundToLong() ==  
Long.MAX_VALUE` when `x > Long.MAX_VALUE`n  
* - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`n */n * @throws  
IllegalArgumentException when this value is `NaN`n */n@SinceKotlin("1.2")npublic expect fun  
Double.roundToLong(): Longn/n// endregionn/n/n// region ===== Float Math  
=====n/n/** Computes the sine of the angle [x] given in  
radians.n */n * Special cases:n * - `sin(NaN|+Inf|-Inf)` is `NaN`n */n@SinceKotlin("1.2")npublic expect fun  
sin(x: Float): Floatn/n/** Computes the cosine of the angle [x] given in radians.n */n * Special cases:n * -  
`cos(NaN|+Inf|-Inf)` is `NaN`n */n@SinceKotlin("1.2")npublic expect fun cos(x: Float): Floatn/n/** Computes  
the tangent of the angle [x] given in radians.n */n * Special cases:n * - `tan(NaN|+Inf|-Inf)` is `NaN`n  
*/n@SinceKotlin("1.2")npublic expect fun tan(x: Float): Floatn/n/**n * Computes the arc sine of the value  
[x];n * the returned value is an angle in the range from  
`-PI/2` to `PI/2` radians.n */n * Special cases:n * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`n  
*/n@SinceKotlin("1.2")npublic expect fun asin(x: Float): Floatn/n/**n * Computes the arc cosine of the value  
[x];n * the returned value is an angle in the range from `0.0` to `PI` radians.n */n * Special cases:n * - `acos(x)`  
is `NaN`, when `abs(x) > 1` or x is `NaN`n */n@SinceKotlin("1.2")npublic expect fun acos(x: Float):  
Floatn/n/**n * Computes the arc tangent of the value [x];n * the returned value is an angle in the range from `  
PI/2` to `PI/2` radians.n */n * Special cases:n * - `atan(NaN)` is `NaN`n */n@SinceKotlin("1.2")npublic  
expect fun atan(x: Float): Floatn/n/**n * Returns the angle `theta` of the polar coordinates `(r, theta)` that  
correspondn * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];n * the`

returned value is an angle in the range from  $-\pi$  to  $\pi$  radians.

**Special cases:**

- $\text{atan2}(0.0, 0.0)$  is  $0.0$
- $\text{atan2}(0.0, x)$  is  $0.0$  for  $x > 0$  and  $\pi$  for  $x < 0$
- $\text{atan2}(-0.0, x)$  is  $-0.0$  for  $x > 0$  and  $-\pi$  for  $x < 0$
- $\text{atan2}(y, +\text{Inf})$  is  $0.0$  for  $0 < y < +\text{Inf}$  and  $-0.0$  for  $-\text{Inf} < y < 0$
- $\text{atan2}(y, -\text{Inf})$  is  $\pi$  for  $0 < y < +\text{Inf}$  and  $-\pi$  for  $-\text{Inf} < y < 0$
- $\text{atan2}(y, 0.0)$  is  $\pi/2$  for  $y > 0$  and  $-\pi/2$  for  $y < 0$
- $\text{atan2}(+\text{Inf}, x)$  is  $\pi/2$  for finite  $x \geq 0$
- $\text{atan2}(-\text{Inf}, x)$  is  $-\pi/2$  for finite  $x \leq 0$
- $\text{atan2}(\text{NaN}, x)$  and  $\text{atan2}(y, \text{NaN})$  is  $\text{NaN}$

```

@SinceKotlin("1.2")
public expect fun atan2(y: Float, x: Float): Float
/**
 * Computes the hyperbolic sine of the value [x].
 *
 * Special cases:
 * - sinh(NaN) is NaN
 * - sinh(+Inf) is +Inf
 * - sinh(-Inf) is -Inf
 */
@SinceKotlin("1.2")
public expect fun sinh(x: Float): Float
/**
 * Computes the hyperbolic cosine of the value [x].
 *
 * Special cases:
 * - cosh(NaN) is NaN
 * - cosh(+Inf) is +Inf
 */
@SinceKotlin("1.2")
public expect fun cosh(x: Float): Float
/**
 * Computes the hyperbolic tangent of the value [x].
 *
 * Special cases:
 * - tanh(NaN) is NaN
 * - tanh(+Inf) is 1.0
 * - tanh(-Inf) is -1.0
 */
@SinceKotlin("1.2")
public expect fun tanh(x: Float): Float
/**
 * Computes the inverse hyperbolic sine of the value [x].
 *
 * The returned value is `y` such that `sinh(y) == x`.
 *
 * Special cases:
 * - asinh(NaN) is NaN
 * - asinh(+Inf) is +Inf
 * - asinh(-Inf) is -Inf
 */
@SinceKotlin("1.2")
public expect fun asinh(x: Float): Float
/**
 * Computes the inverse hyperbolic cosine of the value [x].
 *
 * The returned value is positive `y` such that `cosh(y) == x`.
 *
 * Special cases:
 * - acosh(NaN) is NaN
 * - acosh(x) is NaN when `x < 1`
 * - acosh(+Inf) is +Inf
 */
@SinceKotlin("1.2")
public expect fun acosh(x: Float): Float
/**
 * Computes the inverse hyperbolic tangent of the value [x].
 *
 * The returned value is `y` such that `tanh(y) == x`.
 *
 * Special cases:
 * - tanh(NaN) is NaN
 * - tanh(x) is NaN when `x > 1` or `x < -1`
 * - tanh(1.0) is +Inf
 * - tanh(-1.0) is -Inf
 */
@SinceKotlin("1.2")
public expect fun atanh(x: Float): Float
/**
 * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.
 *
 * Special cases:
 * - returns +Inf if any of arguments is infinite
 * - returns NaN if any of arguments is NaN and the other is not infinite
 */
@SinceKotlin("1.2")
public expect fun hypot(x: Float, y: Float): Float
/**
 * Computes the positive square root of the value [x].
 *
 * Special cases:
 * - sqrt(x) is NaN when `x < 0` or `x` is NaN
 */
@SinceKotlin("1.2")
public expect fun sqrt(x: Float): Float
/**
 * Computes Euler's number `e` raised to the power of the value [x].
 *
 * Special cases:
 * - exp(NaN) is NaN
 * - exp(+Inf) is +Inf
 * - exp(-Inf) is 0.0
 */
@SinceKotlin("1.2")
public expect fun exp(x: Float): Float
/**
 * Computes `exp(x) - 1`.
 *
 * This function can be implemented to produce more precise result for [x] near zero.
 *
 * Special cases:
 * - expm1(NaN) is NaN
 * - expm1(+Inf) is +Inf
 * - expm1(-Inf) is -1.0
 * @see [exp] function.
 */
@SinceKotlin("1.2")
public expect fun expm1(x: Float): Float
/**
 * Computes the logarithm of the value [x] to the given [base].
 *
 * Special cases:
 * - log(x, b) is NaN if either `x` or `b` are NaN
 * - log(x, b) is NaN when `x < 0` or `b <= 0` or `b == 1.0`
 * - log(+Inf, +Inf) is NaN
 * - log(+Inf, b) is +Inf for `b > 1` and -Inf for `b < 1`
 * - log(0.0, b) is -Inf for `b > 1` and +Inf for `b < 1`
 * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].
 */
@SinceKotlin("1.2")
public expect fun log(x: Float, base: Float): Float
/**
 * Computes the natural logarithm (base `E`) of the value [x].
 *
 * Special cases:
 * - ln(NaN) is NaN
 * - ln(x) is NaN when `x < 0.0`
 * - ln(+Inf) is +Inf
 * - ln(0.0) is -Inf
 */
@SinceKotlin("1.2")
public expect fun ln(x: Float): Float
/**
 * Computes the common logarithm (base 10) of the value [x].
 *
 * @see [ln] function for special cases.
 */
@SinceKotlin("1.2")
public expect fun log10(x: Float): Float
/**
 * Computes the binary logarithm (base 2) of the value [x].
 *
 * @see [ln] function for special cases.
 */
@SinceKotlin("1.2")
public expect fun log2(x: Float): Float
/**
 * Computes `ln(a + 1)`.
 *
 * This function can be implemented to produce more precise result for [x] near zero.
 *
 * Special cases:
 * - ln1p(NaN) is NaN
 * - ln1p(x) is NaN where `x < -1.0`
 * - ln1p(-1.0) is -Inf
 * - ln1p(+Inf) is +Inf
 * @see [ln] function
 * @see [expm1] function
 */
@SinceKotlin("1.2")
public expect fun ln1p(x: Float): Float
/**
 * Rounds the given value [x] to an integer towards positive infinity.
 *
 * @return the smallest Float value that is greater than
 */

```

or equal to the given value [x] and is a mathematical integer.  
Special cases: `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
@SinceKotlin("1.2")\npublic expect fun ceil(x: Float): Float  
Rounds the given value [x] to an integer towards negative infinity.  
@return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.  
Special cases:  
`floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
@SinceKotlin("1.2")\npublic expect fun floor(x: Float): Float  
Rounds the given value [x] to an integer towards zero.  
@return the value [x] having its fractional part truncated.  
Special cases:  
`truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
@SinceKotlin("1.2")\npublic expect fun truncate(x: Float): Float  
Rounds the given value [x] towards the closest integer with ties rounded towards even integer.  
Special cases:  
`round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
@SinceKotlin("1.2")\npublic expect fun round(x: Float): Float  
Returns the absolute value of the given value [x].  
Special cases:  
`abs(NaN)` is `NaN`  
@see absoluteValue extension property for [Float]  
@SinceKotlin("1.2")\npublic expect fun abs(x: Float): Float  
Returns the sign of the given value [x]:  
`-1.0` if the value is negative,  
`0` if the value is zero,  
`1.0` if the value is positive  
Special case:  
`sign(NaN)` is `NaN`  
@SinceKotlin("1.2")\npublic expect fun sign(x: Float): Float  
Returns the smaller of two values.  
If either value is `NaN`, then the result is `NaN`.  
@SinceKotlin("1.2")\npublic expect fun min(a: Float, b: Float): Float  
Returns the greater of two values.  
If either value is `NaN`, then the result is `NaN`.  
@SinceKotlin("1.2")\npublic expect fun max(a: Float, b: Float): Float  
// extensions  
Raises this value to the power [x].  
Special cases:  
`b.pow(0.0)` is `1.0`  
`b.pow(1.0) == b`  
`b.pow(NaN)` is `NaN`  
`NaN.pow(x)` is `NaN` for `x != 0.0`  
`b.pow(Inf)` is `NaN` for `abs(b) == 1.0`  
`b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer  
@SinceKotlin("1.2")\npublic expect fun Float.pow(x: Float): Float  
Raises this value to the integer power [n].  
See the other overload of [pow] for details.  
@SinceKotlin("1.2")\npublic expect fun Float.pow(n: Int): Float  
Returns the absolute value of this value.  
Special cases:  
`NaN.absoluteValue` is `NaN`  
@see abs function  
@SinceKotlin("1.2")\npublic expect val Float.absoluteValue: Float  
Returns the sign of this value:  
`-1.0` if the value is negative,  
`0` if the value is zero,  
`1.0` if the value is positive  
Special case:  
`NaN.sign` is `NaN`  
@SinceKotlin("1.2")\npublic expect val Float.sign: Float  
Returns this value with the sign bit same as of the [sign] value.  
If [sign] is `NaN` the sign of the result is undefined.  
@SinceKotlin("1.2")\npublic expect fun Float.withSign(sign: Float): Float  
Returns this value with the sign bit same as of the [sign] value.  
@SinceKotlin("1.2")\npublic expect fun Float.withSign(sign: Int): Float  
Rounds this [Float] value to the nearest integer and converts the result to [Int].  
Ties are rounded towards positive infinity.  
Special cases:  
`x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`  
`x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`  
@throws IllegalArgumentException when this value is `NaN`  
@SinceKotlin("1.2")\npublic expect fun Float.roundToInt(): Int  
Rounds this [Float] value to the nearest integer and converts the result to [Long].  
Ties are rounded towards positive infinity.  
Special cases:  
`x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`  
`x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`  
@throws IllegalArgumentException when this value is `NaN`  
@SinceKotlin("1.2")\npublic expect fun Float.roundToLong(): Long  
// endregion  
// Integer Math  
Returns the absolute value of the given value [n].  
Special cases:  
`abs(Int.MIN_VALUE)` is `Int.MIN_VALUE`  
due to an overflow  
@see absoluteValue extension property for [Int]  
@SinceKotlin("1.2")\npublic expect fun abs(n: Int): Int  
Returns the smaller of two values.  
@SinceKotlin("1.2")\npublic expect

```

fun min(a: Int, b: Int): Int\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin("1.2")\npublic expect
fun max(a: Int, b: Int): Int\n\n/**\n * Returns the absolute value of this value.\n */\n * Special cases:\n * -
`Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n */\n * @see abs function\n
*/\n@SinceKotlin("1.2")\npublic expect val Int.absoluteValue: Int\n\n/**\n * Returns the sign of this value:\n * -
`-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n
*/\n@SinceKotlin("1.2")\npublic expect val Int.sign: Int\n\n\n/**\n * Returns the absolute value of the given
value [n].\n */\n * Special cases:\n * - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow\n
*/\n * @see absoluteValue extension property for [Long]\n */\n@SinceKotlin("1.2")\npublic expect fun abs(n:
Long): Long\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin("1.2")\npublic expect fun min(a:
Long, b: Long): Long\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin("1.2")\npublic expect fun
max(a: Long, b: Long): Long\n\n/**\n * Returns the absolute value of this value.\n */\n * Special cases:\n * -
`Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n */\n * @see abs function\n
*/\n@SinceKotlin("1.2")\npublic expect val Long.absoluteValue: Long\n\n/**\n * Returns the sign of this value:\n
*/\n * - `-1` if the value is negative,\n */\n * - `0` if the value is zero,\n */\n * - `1` if the value is positive\n
*/\n@SinceKotlin("1.2")\npublic expect val Long.sign: Int\n\n\n//
endregion\n\n", "names": [], "mappings": "AAWC,CAXA,yB;EACG,IAAI,OAAO,MAAO,KAAl,UAAW,IAAG,MAA
M,IAAI,C,C;IACI,MAAM,CAAC,QAAD,EAAW,CAAC,SAAD,CAAX,EAawB,OAAxB,C;SAEL,IAAI,OAAO,O
AAQ,KAAl,QAAvB,C;IACD,OAAO,CAAC,MAAM,QAAP,C;;IAGP,IAAI,OAAQ,GAAE,E;IACd,OAAO,CAAC
,IAAI,OAAL,C;;CAEd,CAAC,IAAD,EAAO,kB;EACJ,IAAI,IAAI,M;ECPZ,MAAM,eAAgB,GAAE,a;IACpB,OA
AoD,CAA5C,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAE,YAAW,SAAW,KAAG,CAAC,OAAQ,KAAl,c;G
;EAGxE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,SAAU,IAAG,CAAC,OAAQ,KAAl,c;G;EAGID,M
AAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,U;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,Y
AAW,WAAY,IAAG,CAAC,OAAQ,KAAl,W;G;EAGpD,MAAM,WAAY,GAAE,a;IACbB,OAAO,CAAE,YAAW,
U;G;EAGxB,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,Y;G;EAGxB,MAAM,cAAe,GAAE,a;IACnB,O
AAO,CAAE,YAAW,Y;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,IA
AG,CAAC,OAAQ,KAAl,W;G;EAG5C,MAAM,QAAS,GAAE,a;IACb,OAAO,KAAK,QAAQ,CAAC,CAAD,CAA
L,IAAG,CAAC,CAAC,O;G;EAGjC,MAAM,WAAY,GAAE,a;IACbB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,
IAAG,WAAW,OAAO,CAAC,CAAD,C;G;EAGjD,MAAM,cAAe,GAAE,a;IACnB,IAAI,CAAE,KAAl,IAAV,C;M
AAgB,OAAO,M;IACvB,IAAI,WAAW,MAAM,YAAY,CAAC,CAAD,CAAI,GAAE,MAAM,aAAR,GAAwB,MA
AM,S;IACnE,OAAO,GAAl,GAAE,KAAK,UAAU,IAAI,KAAK,CAAC,CAAD,EAAl,a;MAAc,OAAO,QAAQ,CA
AC,CAAD,C;KAAjC,CAAwC,KAAK,CAAC,IAAD,CAAQ,GAAE,G;G;EAG/F,MAAM,kBAAmB,GAAE,e;IACv
B,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G;EAG5D,MAAM,YAAa,GAAE,gB;IACjB,IAAI,C
AAE,KAAl,CAAV,C;MACI,OAAO,I;;IAEX,IAAI,CAAE,KAAl,IAAK,IAAG,CAAE,KAAl,IAAK,IAAG,CAAC,
MAAM,WAAW,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAl,CAAC,OAAvE,C;MACI,OAAO,K;;IAGX,KA
AK,IAAI,IAAI,CAAR,EAAW,IAAI,CAAC,OAArB,EAAsB,CAAE,GAAE,CAAIC,EAaqC,CAAC,EAAtC,C;MA
CI,IAAI,CAAC,MAAM,OAAO,CAAC,CAAC,CAAC,CAAD,CAAF,EAAsB,CAAC,CAAC,CAAD,CAAR,CAAIB
,C;QACI,OAAO,K;;IAGf,OAAO,I;G;EAGX,MAAM,gBAAiB,GAAE,gB;IACrB,OAAO,MAAM,OAAO,YAAY,s
BAAsB,CAAC,CAAD,EAAl,CAAJ,C;G;EAGID,MAAM,cAAe,GAAE,e;IACnB,IAAI,GAAl,KAAl,IAAZ,C;MA
AkB,OAAO,C;IACzB,IAAI,SAAS,C;IACb,KAAK,IAAI,IAAI,CAAR,EAAW,IAAI,GAAG,OAAvB,EAAGC,CAA
E,GAAE,CAApC,EAAsC,CAAC,EAAX,C,C;MACI,MAAO,GAAqB,CAAjB,EAAG,GAAE,MAAO,GAAE,CAAG
,IAAE,MAAM,SAAS,CAAC,GAAG,CAAC,CAAD,CAAJ,CAAU,GAAE,C;;IAE7D,OAAO,M;G;EAGX,MAAM,
kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G;EAG5D,MAAM,mBAA
oB,GAAE,iB;IACxB,KAAK,KAAK,CAAC,MAAM,gBAAP,C;G;ECPFd,MAAM,eAAgB,GAAE,mB;IACpB,CA
AC,aAAc,GAAE,I;IACjB,OAAO,C;G;EAGX,MAAM,uBAAwB,GAAE,4C;IAC5B,MAAM,IAAK,GAAE,M;IAC
b,MAAM,IAAK,GAAE,M;IACb,MAAM,aAAc,GAAE,I;IACtB,OAAO,mBAAmB,CAAC,MAAD,EAAS,MAAT,
EAAl,6BAA6B,CAAC,UAAD,CAA9C,C;G;EAG9B,iD;IACI,GAAG,WAAY,GAAE,sBAAsB,CAAC,OAAO,M
AAO,KAAl,UAAW,GAAE,KAAK,QAAP,GAakB,KAAK,UAArD,C;IACvC,GAAG,YAAa,GAAE,G;IACIB,OA
AO,G;G;EAGX,IAAI,gCAAgC,CACcC,UACa,QAAS,IAAT,wBAAqC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,k

```

B;GADvB,CADb,aAIe,QAAS,IAAT,wBAAqC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CADgC ,EAShC,UACa,QAAS,IAAT,wBAAqC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,kB;GADvB,CADb,aAIe,QAAS,I AAT,wBAAqC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CATgC,C;EAmBpC,uC;IACI,IAAI,KA AK,MAAO,KAAI,IAApB,C;MACI,KAAK,MAAO,GAAE,aACE,CAAC,KAAK,qBAAqB,EAA3B,CADF,aAEC,I AFD,aAGC,EAHD,cAIE,EAJF,SAKH,EALG,iBAMK,EANL,C;;IASIB,OAAO,KAAK,M;G;EChDhB,MAAM,QA AS,GAAE,a;IACb,OAAoB,CAAZ,CAAE,GAAE,KAAQ,KAAG,EAAG,IAAG,E;G;EAGjC,MAAM,OAAQ,GAA E,a;IACZ,OAAkB,CAAV,CAAE,GAAE,GAAM,KAAG,EAAG,IAAG,E;G;EAG/B,MAAM,OAAQ,GAAE,a;IAC Z,OAAO,CAAE,GAAE,K;G;EAGf,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE, CAAF,GAAM,MAAM,KAAK,WAAW,CAAC,CAAD,C;G;EAGhE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE ,YAAW,MAAM,KAAM,GAAE,CAAC,MAAM,EAAT,GAAC,MAAM,YAAY,CAAC,CAAD,C;G;EAGpE,MAA M,cAAe,GAAE,a;IACnB,OAAO,MAAM,QAAQ,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGzB,MA AM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,M AAM,eAAgB,GAAE,a;IACpB,OAAO,CAAC,C;G;EAGZ,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO, CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,GAAE,U AAR,C;MAAoB,OAAO,U;IAC3B,IAAI,CAAE,GAAE,WAAR,C;MAAqB,OAAO,W;IAC5B,OAAO,CAAE,GAA E,C;G;EAGf,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,IAAI,CAAE,Y AAW,MAAM,UAAvB,C;MAAmC,OAAO,C;IAC1C,OAAO,IAAI,MAAM,UAAV,CAAqB,CAArB,C;G;EAGX,M AAM,UAAW,GAAE,a;IACf,IAAI,CAAE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,OAAO,MAAM,OAAO,CAAC, CAAD,C;G;ECIDxB,MAAM,OAAQ,GAAE,sB;IACZ,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,IAAK,IAAG,I;I AGnB,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,K;;IAGX,IAAI,IAAK,KAAI,IAAb,C;MACI,OAAO,IAAK,KAA LI;;IAGpB,IAAI,OAAO,IAAK,KAAI,QAAS,IAAG,OAAO,IAAI,OAAQ,KAAI,UAAvD,C;MACI,OAAO,IAAI,O AAO,CAAC,IAAD,C;;IAGtB,IAAI,OAAO,IAAK,KAAI,QAAS,IAAG,OAAO,IAAK,KAAI,QAAdD,C;MACI,OA AO,IAAK,KAAI,IAAK,KAAI,IAAK,KAAI,CAAE,IAAG,CAAE,GAAE,IAAK,KAAI,CAAE,GAAE,IAAnC,C;;I AGzB,OAAO,IAAK,KAAI,I;G;EAGpB,MAAM,SAAU,GAAE,e;IACd,IAAI,GAAI,IAAG,IAAX,C;MACI,OAAO, C;;IAEX,IAAI,UAAU,OAAO,G;IACrB,IAAI,QAAS,KAAI,OAAjB,C;MACI,OAAO,UAAW,KAAI,OAAO,GAA G,SAAU,GAAE,GAAG,SAAS,EAAd,GAAMb,iBAAiB,CAAC,GAAD,C;;IAEIF,IAAI,UAAW,KAAI,OAAnB,C; MACI,OAAO,iBAAiB,CAAC,GAAD,C;;IAE5B,IAAI,QAAS,KAAI,OAAjB,C;MACI,OAAO,MAAM,eAAe,CAA C,GAAD,C;;IAEhC,IAAI,SAAU,KAAI,OAaIB,C;MACI,OAAO,MAAM,CAAC,GAAD,C;;IAGjB,IAAI,MAAM, MAAM,CAAC,GAAD,C;IACHb,OAAO,iBAAiB,CAAC,GAAD,C;G;EAI5B,MAAM,SAAU,GAAE,a;IACd,IAAI, CAAE,IAAG,IAAT,C;MACI,OAAO,M;WAEN,IAAI,MAAM,WAAW,CAAC,CAAD,CAArB,C;MACD,OAAO,O ;;MAGP,OAAO,CAAC,SAAS,E;G;EAKzB,IAAI,WAAW,a;EAGf,IAAI,iCAAiC,sB;EAErC,gC;IACI,IAAI,EAAE ,8BAA+B,IAAG,GAApC,CAAJ,C;MACI,IAAI,OAAQ,IAAI,OAAO,EAAG,GAAE,QAAU,GAAE,C;MACxC,MA AM,eAAe,CAAC,GAAD,EAAM,8BAAN,EAAc,QAAU,IAAV,cAA4B,KAA5B,CAAtC,C;;IAEzB,OAAO,GAA G,CAAC,8BAAD,C;G;EAGd,gC;IACL,IAAI,OAAO,C;IACX,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,G AAG,OAAvB,EAAGC,CAAC,EAajC,C;MACI,IAAI,OAAQ,GAAG,WAAW,CAAC,CAAD,C;MAC1B,IAAM,G AAG,IAAK,GAAE,EAAG,GAAE,IAAM,GAAE,C;;IAEjC,OAAO,I;G;EAGX,MAAM,iBAakB,GAAE,iB;EC9C1 B,MAAM,KAAM,GAAE,qB;IAKZ,IAAI,KAAM,GAAE,GAAL,GAAE,C;IAMIB,IAAI,MAAO,GAAE,IAAK,GA AE,C;G;EAGtB,MAAM,KAAK,WAAW,GAAE,OAcF,OADe,cAET,MAFS,cAGV,EAHU,C;EAgBzB,MAAM,KA AK,UAAW,GAAE,E;EAQxB,MAAM,KAAK,QAAS,GAAE,iB;IACpB,IAAI,IAAK,IAAG,KAAM,IAAG,KAAM, GAAE,GAA7B,C;MACE,IAAI,YAAY,MAAM,KAAK,UAAU,CAAC,KAAD,C;MACrC,IAAI,SAAJ,C;QACE,O AAO,S;;IAIX,IAAI,MAAM,IAAI,MAAM,KAAV,CAAGB,KAAM,GAAE,CAAxB,EAA2B,KAAM,GAAE,CAA E,GAAE,EAAF,GAAO,CAA5C,C;IACV,IAAI,IAAK,IAAG,KAAM,IAAG,KAAM,GAAE,GAA7B,C;MACE,MA AM,KAAK,UAAU,CAAC,KAAD,CAAQ,GAAE,G;;IAEjC,OAAO,G;G;EAYT,MAAM,KAAK,WAAW,GAAE,iB ;IACvB,IAAI,KAAK,CAAC,KAAD,CAAT,C;MACE,OAAO,MAAM,KAAK,K;WACb,IAAI,KAAM,IAAG,CAA C,MAAM,KAAK,gBAAzB,C;MACL,OAAO,MAAM,KAAK,U;WACb,IAAI,KAAM,GAAE,CAAE,IAAG,MAA M,KAAK,gBAA5B,C;MACL,OAAO,MAAM,KAAK,U;WACb,IAAI,KAAM,GAAE,CAAZ,C;MACL,OAAO,MA AM,KAAK,WAAW,CAAC,CAAC,KAAF,CAAQ,OAAO,E;;MAE5C,OAAO,IAAI,MAAM,KAAV,CACF,KAAM ,GAAE,MAAM,KAAK,gBAakB,GAAE,CADrC,EAef,KAAM,GAAE,MAAM,KAAK,gBAakB,GAAE,CAFrC,

C;;G;EAcX,MAAM,KAAK,SAAU,GAAE,6B;IACrB,OAAO,IAAI,MAAM,KAAV,CAAgB,OAAhB,EAAyB,QA  
AzB,C;G;EAWT,MAAM,KAAK,WAAY,GAAE,0B;IACvB,IAAI,GAAG,OAAQ,IAAG,CAAIB,C;MACE,MAAM  
,KAAK,CAAC,mCAAD,C;;IAGb,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE,CAAIE,IAAG,EAAG,  
GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAA1B,C;;IAGb,IAAI,GAAG,OAAO,CAAC,C  
AAD,CAAI,IAAG,GAARb,C;MACE,OAAO,MAAM,KAAK,WAAW,CAAC,GAAG,UAAU,CAAC,CAAD,CAAd  
,EAAmB,KAAmB,CAAyB,OAAO,E;WACxD,IAAI,GAAG,QAAQ,CAAC,GAAD,CAAM,IAAG,CAAxB,C;MAC  
L,MAAM,KAAK,CAAC,+CAAgD,GAAE,GAAnD,C;;IAKb,IAAI,eAAe,MAAM,KAAK,WAAW,CAAC,IAAI,IA  
AI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,SAAS,MAAM,KAAK,K;IACxB,KAAK,IAAI,IAAI,CAA  
b,EAAgB,CAAIE,GAAE,GAAG,OAAvB,EAAGC,CAAIE,IAAG,CAArC,C;MACE,IAAI,OAAO,IAAI,IAAI,CAAC  
,CAAD,EAAI,GAAG,OAAQ,GAAE,CAAjB,C;MACnB,IAAI,QAAQ,QAAQ,CAAC,GAAG,UAAU,CAAC,CAA  
D,EAAI,CAAIE,GAAE,IAAR,CAAd,EAA6B,KAA7B,C;MACpB,IAAI,IAAK,GAAE,CAAX,C;QACE,IAAI,QAA  
Q,MAAM,KAAK,WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,EAAQ,IAAR,CAAT,C;QACIC,MAAO,GAAE,MAA  
M,SAAS,CAAC,KAAD,CAAO,IAAI,CAAC,MAAM,KAAK,WAAW,CAAC,KAAD,CAAvB,C;;QAEhC,MAAO,  
GAAE,MAAM,SAAS,CAAC,YAAD,C;QACxB,MAAO,GAAE,MAAM,IAAI,CAAC,MAAM,KAAK,WAAW,CA  
AC,KAAD,CAAvB,C;;IAGvB,OAAO,M;G;EAcT,MAAM,KAAK,gBAAiB,GAAE,CAAIE,IAAG,E;EAOnc,MA  
AM,KAAK,gBAAiB,GAAE,CAAIE,IAAG,E;EAOnc,MAAM,KAAK,gBAAiB,GACxB,MAAM,KAAK,gBAAiB,  
GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gBAAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,C;EAOIC,  
MAAM,KAAK,gBAAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gB  
AAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gBAAiB,GACxB,MA  
AM,KAAK,gBAAiB,GAAE,C;EAIIC,MAAM,KAAK,KAAM,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAD,C;EA  
ItC,MAAM,KAAK,IAAK,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAD,C;EAIrC,MAAM,KAAK,QAAS,GAAE,  
MAAM,KAAK,QAAQ,CAAC,EAAD,C;EAIzC,MAAM,KAAK,UAAW,GACIB,MAAM,KAAK,SAAS,CAAC,aA  
AW,GAAE,CAAd,EAAiB,UAAW,GAAE,CAA9B,C;EAIxB,MAAM,KAAK,UAAW,GAAE,MAAM,KAAK,SAA  
S,CAAC,CAAD,EAAI,aAAW,GAAE,CAAjB,C;EAO5C,MAAM,KAAK,YAAa,GAAE,MAAM,KAAK,QAAQ,C  
AAC,CAAIE,IAAG,EAAN,C;EAI7C,MAAM,KAAK,UAAU,MAAO,GAAE,Y;IAC5B,OAAO,IAAI,K;G;EAKb,M  
AAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,MAAM,KAAK,gBAAiB,GACzC,IAAI  
,mBAAmB,E;G;EAIhC,MAAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,IAAI,K;G;E  
AQ1B,MAAM,KAAK,UAAU,SAAU,GAAE,qB;IAC/B,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE  
,CAAIE,IAAG,EAAG,GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAA1B,C;;IAGb,IAAI,I  
AAI,OAAO,EAAf,C;MACE,OAAO,G;;IAGT,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,CAAC,  
MAAM,KAAK,UAAZ,CAAnB,C;QAGE,IAAI,YAAY,MAAM,KAAK,WAAW,CAAC,KAAD,C;QACIC,IAAI,M  
AAM,IAAI,IAAI,CAAC,SAAD,C;QACIB,IAAI,MAAM,GAAG,SAAS,CAAC,SAAD,CAAW,SAAS,CAAC,IAA  
D,C;QACIC,OAAO,GAAG,SAAS,CAAC,KAAD,CAAQ,GAAE,GAAG,MAAM,EAAE,SAAS,CAAC,KAAD,C;;  
QAEjD,OAAO,GAAI,GAAE,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,C;;IAMvC,IAAI,eAAe,MAAM,KAAK,  
WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,MAAM,I;IACV,IAAI,SAAS,E;IA  
Cb,OAAO,IAAP,C;MACE,IAAI,SAAS,GAAG,IAAI,CAAC,YAAD,C;MACpB,IAAI,SAAS,GAAG,SAAS,CAAC  
,MAAM,SAAS,CAAC,YAAD,CAAhB,CAA+B,MAAM,E;MAC9D,IAAI,SAAS,MAAM,SAAS,CAAC,KAAD,C;  
MAE5B,GAAL,GAAE,M;MACN,IAAI,GAAG,OAAO,EAAd,C;QACE,OAAO,MAAO,GAAE,M;;QAEhB,OAAO,  
MAAM,OAAQ,GAAE,CAAvB,C;UACE,MAAO,GAAE,GAAL,GAAE,M;;QAEjB,MAAO,GAAE,EAAG,GAAE,  
MAAO,GAAE,M;;G;EAO7B,MAAM,KAAK,UAAU,YAAa,GAAE,Y;IACIC,OAAO,IAAI,M;G;EAKb,MAAM,K  
AAK,UAAU,WAAY,GAAE,Y;IACjC,OAAO,IAAI,K;G;EAKb,MAAM,KAAK,UAAU,mBAAoB,GAAE,Y;IACz  
C,OAAQ,IAAI,KAAM,IAAG,CAAG,GACpB,IAAI,KADgB,GACR,MAAM,KAAK,gBAAiB,GAAE,IAAI,K;G;E  
AQpD,MAAM,KAAK,UAAU,cAAe,GAAE,Y;IACpC,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,  
CAAC,MAAM,KAAK,UAAZ,CAAnB,C;QACE,OAAO,E;;QAEp,OAAO,IAAI,OAAO,EAAE,cAAc,E;;MAGpC,  
IAAI,MAAM,IAAI,MAAO,IAAG,CAAIE,GAAE,IAAI,MAAN,GAAe,IAAI,K;MAC7C,KAAK,IAAI,MAAM,EA  
Af,EAAmB,GAAL,GAAE,CAAzB,EAA4B,GAAG,EAA/B,C;QACE,IAAuB,CAAIB,GAAL,GAAG,CAAIE,IAAG,  
GAAM,KAAG,CAA1B,C;UACE,K;;MAGJ,OAAO,IAAI,MAAO,IAAG,CAAIE,GAAE,GAAL,GAAE,EAAR,GA  
Aa,GAAL,GAAE,C;;G;EAM9C,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,OAAO,IAAI,MAAO,IAAG,CAA

E,IAAG,IAAI,KAAM,IAAG,C;G;EAKzC,MAAM,KAAK,UAAU,WAAY,GAAE,Y;IACjC,OAAO,IAAI,MAAO,  
GAAE,C;G;EAKtB,MAAM,KAAK,UAAU,MAAO,GAAE,Y;IAC5B,OAAuB,CAAf,IAAI,KAAM,GAAE,CAAG,  
KAAG,C;G;EAQ5B,MAAM,KAAK,UAAU,WAAY,GAAE,iB;IACjC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ  
,IAAI,IAAI,KAAM,IAAG,KAAK,K;G;EAQ3D,MAAM,KAAK,UAAU,cAAe,GAAE,iB;IACpC,OAAQ,IAAI,MA  
AO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAAM,IAAG,KAAK,K;G;EAQ3D,MAAM,KAAK,UAAU,SAAU,GAAE,i  
B;IAC/B,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;G;EAQ/B,MAAM,KAAK,UAAU,gBAAiB,GAAE,  
iB;IACtC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;G;EAQhC,MAAM,KAAK,UAAU,YAAa,GAAE,iB  
;IACiC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;G;EAQ/B,MAAM,KAAK,UAAU,mBAaOB,GAAE,i  
B;IACzC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;G;EAUhC,MAAM,KAAK,UAAU,QAAS,GAAE,iB  
;IAC9B,IAAI,IAAI,WAAW,CAAC,KAAD,CAAnB,C;MACE,OAAO,C;;IAGT,IAAI,UAAU,IAAI,WAAW,E;IAC  
7B,IAAI,WAAW,KAAK,WAAW,E;IAC/B,IAAI,OAAQ,IAAG,CAAC,QAahB,C;MACE,OAAO,E;;IAET,IAAI,C  
AAC,OAAQ,IAAG,QAahB,C;MACE,OAAO,C;;IAIT,IAAI,IAAI,SAAS,CAAC,KAAD,CAAO,WAAW,EAAnC,  
C;MACE,OAAO,E;;MAEP,OAAO,C;;G;EAMX,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,IAAI,IAAI,WA  
AW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,MAAM,KAAK,U;;MAEiB,OAAO,IAAI,IAAI,EA  
E,IAAI,CAAC,MAAM,KAAK,IAAZ,C;;G;EAUzB,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAG1B,IAAI,MAAM  
,IAAI,MAAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,IAAI,KAAM,KAAI,E;I  
ACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MAAO,KAAI,E;IAC1B,IAAI,MAAM,K  
AAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB,IAAI,MAAM,KAAK,KAAM,GAA  
E,K;IAEvB,IAAI,MAAM,CAAV,EAaA,MAAM,CAAnB,EAAsB,MAAM,CAA5B,EAa+B,MAAM,C;IACrC,GA  
AI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GA  
AE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,I  
AAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,K;IACP,OAAO  
,MAAM,KAAK,SAAS,CAAE,GAAI,IAAG,EAAL,GAAE,GAaf,EAaqB,GAAI,IAAG,EAAL,GAAE,GAaIC,C;G;  
EAS7B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,IAAI,CAAC,KAAK,OAAO,EAAb,C;G;EA  
SjB,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,IAAI,IAAI,OAAO,EAaf,C;MACE,OAAO,MAAM,KAAK,K  
;WACb,IAAI,KAAK,OAAO,EAahB,C;MACL,OAAO,MAAM,KAAK,K;;IAGpB,IAAI,IAAI,WAAW,CAAC,MA  
AM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,KAAK,MAAM,EAAG,GAAE,MAAM,KAAK,UAAb,GAA0B,MA  
AM,KAAK,K;WACrD,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK,UAAZ,CAApB,C;MACL,OAAO,IAAI,MA  
AM,EAAG,GAAE,MAAM,KAAK,UAAb,GAA0B,MAAM,KAAK,K;;IAG3D,IAAI,IAAI,WAAW,EAAnB,C;MA  
CE,IAAI,KAAK,WAAW,EAApB,C;QACE,OAAO,IAAI,OAAO,EAaE,SAAS,CAAC,KAAK,OAAO,EAAb,C;;Q  
AE7B,OAAO,IAAI,OAAO,EAaE,SAAS,CAAC,KAAD,CAAO,OAAO,E;;WAExC,IAAI,KAAK,WAAW,EAAPB  
,C;MACL,OAAO,IAAI,SAAS,CAAC,KAAK,OAAO,EAAb,CAAgB,OAAO,E;;IAI7C,IAAI,IAAI,SAAS,CAAC,  
MAAM,KAAK,YAAZ,CAA0B,IACvC,KAAK,SAAS,CAAC,MAAM,KAAK,YAAZ,CADiB,C;MAEE,OAAO,M  
AAM,KAAK,WAAW,CAAC,IAAI,SAAS,EAAG,GAAE,KAAK,SAAS,EAajC,C;;IAM/B,IAAI,MAAM,IAAI,M  
AAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,IAAI,KAAM,KAAI,E;IACxB,I  
AAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MAAO,KAAI,E;IAC1B,IAAI,MAAM,KAAK,  
MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB,IAAI,MAAM,KAAK,KAAM,GAAE,K;I  
AEvB,IAAI,MAAM,CAAV,EAaA,MAAM,CAAnB,EAAsB,MAAM,CAA5B,EAa+B,MAAM,C;IACrC,GAAI,IA  
AG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;I  
ACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,G  
AAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,G  
AAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,  
GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,  
GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,G;IACjD,GAAI,IAA  
G,K;IACP,OAAO,MAAM,KAAK,SAAS,CAAE,GAAI,IAAG,EAAL,GAAE,GAaf,EAaqB,GAAI,IAAG,EAAL,G  
AAE,GAaIC,C;G;EAS7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,IAAI,KAAK,OAAO,EAahB,C;MACE  
,MAAM,KAAK,CAAC,kBAAD,C;WACN,IAAI,IAAI,OAAO,EAaf,C;MACL,OAAO,MAAM,KAAK,K;;IAGpB,  
IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,IAAI,KAAK,WAAW,CAAC,MAAM,KAA



K,IAAZ,CAAkB,IACIC,KAAK,WAAW,CAAC,MAAM,KAAK,QAAZ,CADpB,C;QAE,E,OAAO,MAAM,KAAK,U;aACb,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK,UAAZ,CAApB,C;QACL,OAAO,MAAM,KAAK,I;;QAGIB,IAAI,WAAW,IAAI,WAAW,CAAC,CAAD,C;QAC9B,IAAI,SAAS,QAAQ,IAAI,CAAC,KAAD,CAAO,UAAU,C AAC,CAAD,C;QAC1C,IAAI,MAAM,WAAW,CAAC,MAAM,KAAK,KAAZ,CAArB,C;UACE,OAAO,KAAK,W AAW,EAAG,GAAE,MAAM,KAAK,IAAb,GAAoB,MAAM,KAAK,Q;;UAeZD,IAAI,MAAM,IAAI,SAAS,CAAC, KAAK,SAAS,CAAC,MAAD,CAAf,C;UACvB,IAAI,SAAS,MAAM,IAAI,CAAC,GAAG,IAAI,CAAC,KAAD,CA AR,C;UACvB,OAAO,M;;;WAGN,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK,UAAZ,CAApB,C;MACL,OAAO ,MAAM,KAAK,K;;IAGpB,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,KAAK,WAAW,EAAPB,C;QACE,OAAO, IAAI,OAAO,EAAE,IAAI,CAAC,KAAK,OAAO,EAAb,C;;QAExB,OAAO,IAAI,OAAO,EAAE,IAAI,CAAC,KAA D,CAAO,OAAO,E;;WAEnC,IAAI,KAAK,WAAW,EAAPB,C;MACL,OAAO,IAAI,IAAI,CAAC,KAAK,OAAO,E AAb,CAAgB,OAAO,E;;IAQxC,IAAI,MAAM,MAAM,KAAK,K;IACrB,IAAI,MAAM,I;IACV,OAAO,GAAG,mB AAmB,CAAC,KAAD,CAA7B,C;MAGE,IAAI,SAAS,IAAI,IAAI,CAAC,CAAD,EAAl,IAAI,MAAM,CAAC,GAA G,SAAS,EAAG,GAAE,KAAK,SAAS,EAhC,CAAd,C;MAIrB,IAAI,OAAO,IAAI,KAAK,CAAC,IAAI,IAAI,CA AC,MAAD,CAAS,GAAE,IAAI,IAAxB,C;MACpB,IAAI,QAAS,IAAK,IAAG,EAAl,GAAE,CAAF,GAAM,IAAI,I AAI,CAAC,CAAD,EAAl,IAAK,GAAE,EAAX,C;MAIvC,IAAI,YAAY,MAAM,KAAK,WAAW,CAAC,MAAD,C ;MACtC,IAAI,YAAY,SAAS,SAAS,CAAC,KAAD,C;MACIC,OAAO,SAAS,WAAW,EAAG,IAAG,SAAS,YAAY, CAAC,GAAD,CAAtD,C;QACE,MAAO,IAAG,K;QACV,SAAU,GAAE,MAAM,KAAK,WAAW,CAAC,MAAD,C ;QACIC,SAAU,GAAE,SAAS,SAAS,CAAC,KAAD,C;;MAKhC,IAAI,SAAS,OAAO,EAAPB,C;QACE,SAAU,GA AE,MAAM,KAAK,I;;MAGzB,GAAl,GAAE,GAAG,IAAI,CAAC,SAAD,C;MACb,GAAl,GAAE,GAAG,SAAS,C AAC,SAAD,C;;IAEpB,OAAO,G;G;EAST,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC7B,OAAO,IAAI,SAAS,C AAC,IAAI,IAAI,CAAC,KAAD,CAAO,SAAS,CAAC,KAAD,CAAzB,C;G;EAKtB,MAAM,KAAK,UAAU,IAAK, GAAE,Y;IAC1B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAC,IAAI,KAAK,EAaA,CAAC,IAAI,MAAlB,C;G;EAS 7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KAAK,SAAS,CAAC,IAAI,KAAM,GAAE,K AAK,KAAIB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;G;EAU7B,MAAM,KAAK,UAAU,GAAl,GAAE,iB;I ACzB,OAAO,MAAM,KAAK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAIB,EACI,IAAI,MAAO,GAAE,KA AK,MADtB,C;G;EAU7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KAAK,SAAS,CAAC,I AAI,KAAM,GAAE,KAAK,KAAIB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;G;EAU7B,MAAM,KAAK,UA AU,UAAW,GAAE,mB;IAChC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI ,MAAM,IAAI,K;MACd,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,OAAO,IAAI,M;QACf,OAAO,MAAM,KAAK ,SAAS,CACvB,GAAl,IAAG,OADgB,EAEtB,IAAK,IAAG,OAAS,GAAG,GAAl,KAAK,EAAG,GAAE,OAFZ,C;; QAI3B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAD,EAAl,GAAl,IAAI,OAAQ,GAAE,EAAtB,C;;;G;EAWjC,MA AM,KAAK,UAAU,WAAW,GAAE,mB;IACjC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO, I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO, MAAM,KAAK,SAAS,CACtB,GAAl,KAAI,OAAS,GAAG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,IAA G,OAFc,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CACvB,IAAK,IAAI,OAAQ,GAAE,EADl,EAEvB,IAAK,IAAG ,CAAE,GAAE,CAAF,GAAM,EAFO,C;;;G;EAejC,MAAM,KAAK,UAAU,mBAAoB,GAAE,mB;IACzC,OAAQ,I AAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,G AAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,MAAM,KAAK,SAAS,CACtB,GAAl,KAAI,OAAS,GA AG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,KAAI,OAFc,C;aAGtB,IAAI,OAAQ,IAAG,EAaf,C;QACL, OAAO,MAAM,KAAK,SAAS,CAAC,IAAD,EAAO,CAAP,C;;QAE3B,OAAO,MAAM,KAAK,SAAS,CAAC,IAA K,KAAK,OAAQ,GAAE,EAARb,EAAB,CAA1B,C;;;G;EAMjC,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC3B, OAAO,KAAM,YAAW,MAAM,KAAM,IAAG,IAAI,WAAW,CAAC,KAAD,C;G;EAG1D,MAAM,KAAK,UAAU, gBAAiB,GAAE,MAAM,KAAK,UAAU,Q;EAE7D,MAAM,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,I AAI,CAAC,MAAM,KAAK,IAAZ,C;G;EAGnB,MAAM,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAA I,CAAC,MAAM,KAAK,QAAZ,C;G;EAGnB,MAAM,KAAK,UAAU,QAAS,GAAE,Y;IAC5B,OAAO,IAAI,SAAS ,E;G;EAGxB,MAAM,KAAK,UAAU,UAAW,GAAE,Y;IAC9B,OAAO,I;G;EAGX,MAAM,KAAK,UAAU,WAAW, GAAE,MAAM,KAAK,UAAU,O;EACxD,MAAM,KAAK,UAAU,IAAK,GAAE,MAAM,KAAK,UAAU,I;EAEjD, MAAM,KAAK,UAAU,QAAS,GAAE,iB;IAC5B,OAAO,IAAI,MAAM,OAAO,OAAO,UAAxB,CAAmC,IAAnC,E

AAyC,KAAzC,C;G;EC1zBX,MAAM,aAAc,GAAE,2B;G;EAGtB,MAAM,qBAAsB,GAAE,oB;IAC1B,OAAO,G;  
G;EAGX,MAAM,aAAc,GAAE,e;IAC1B,IAAI,IAAI,Y;MACJ,CAAE,GAAE,GAAG,E;MACP,OAAO,CAAC,MA  
AM,CAAC,IAAD,EAAO,SAAP,C;K;IAE1B,OAAO,Y;MACH,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP  
,C;K;G;EAI7B,MAAM,SAAU,GAAE,gB;IACd,OAAO,kB;MACH,OAAO,OAAO,MAAO,KAAI,I;K;G;EAIjC,MA  
AM,aAAc,GAAE,iB;IAC1B,OAAO,kB;MACH,OAAO,MAAM,OAAO,CAAC,MAAD,EAAS,KAAT,C;K;G;EAI5  
B,MAAM,OAAQ,GAAE,c;IACZ,OAAO,kB;MACH,OAAO,MAAO,IAAG,IAAK,IAAG,EAAE,CAAC,MAAD,C;  
K;G;EAI7C,MAAM,aAAc,GAAE,gB;IAC1B,OAAO,kB;MACH,OAAO,CAAC,CAAC,MAAD,CAAS,IAAG,CAA  
C,CAAC,MAAD,C;K;G;EAI7B,MAAM,qBAAsB,GAAE,wC;G;EAG9B,MAAM,YAAa,GAAE,iB;IACjB,OAAO,  
K;G;EAGX,MAAM,gBAAiB,GAAE,qB;IACrB,gBAAgB,E;G;EAGpB,MAAM,oBAAqB,GAAE,qB;IACzB,gBA  
AgB,E;G;EAGpB,MAAM,kBAAmB,GAAE,qB;IACvB,gBAAgB,E;G;EAGpB,MAAM,mBAAoB,GAAE,4B;IACx  
B,gBAAgB,E;G;EAGpB,MAAM,6BAA8B,GAAE,yB;IACIC,gBAAgB,E;G;EAGpB,4B;IACI,MAAM,IAAI,KA  
AJ,CACF,iDAAKD,GACID,qDAAsD,GACtD,uDAHE,C;G;EAMV,MAAM,gBAAiB,GAAE,4B;IACrB,OAAO,Y;M  
ACH,OAAO,Y;K;G;ECJfF,MAAM,UAAW,GAAE,gB;IACf,IAAI,QAAQ,OAAO,C;IACnB,IAAI,KAAM,KA  
AI,QAAAD,C;MACI,IAAI,OAAO,CAAE,KA  
AI,QAAjB,C;QACI,OAAO,MAAM,gBAAgB,CAAC,CAAD,EAAI,CAAJ,C;;MAEjC,OAAO,MAAM,mBAAmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,IAAI,KAAM,KA  
AI,QAAS,IAAG,KAAM,KA  
AI,SAAP,C;MACI,OAAO,MAAM,mBAAmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,OAAO,CAAC,g  
BAAgB,CAAC,CAAD,C;G;EAG5B,MAAM,mBAAoB,GAAE,gB;IACxB,OAAO,CAAE,GAAE,CAAE,GAAE,E  
AAF,GAAO,CAAE,GAAE,CAAE,GAAE,CAAF,GAAM,C;G;EAGpC,MAAM,gBAAiB,GAAE,gB;IACrB,IAAI,C  
AAE,GAAE,CAAR,C;MAAW,OAAO,E;IAC1B,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,C;IAE1B,IAAI,CA  
AE,KA  
AI,CAAV,C;MACI,IAAI,CAAE,KA  
AI,CAAV,C;QAAa,OAAO,C;MAEpB,IAAI,KA  
AK,CAAE,GAAE,C;MACb,OAAO,EAAG,KA  
AI,CAAE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EAAF,GA  
AO,C;;IAG7C,OAAO,CAAE,KA  
AI,CAAE,GAAG,CAAE,KA  
AI,CAAE,GAAE,CAAF,GAAM,CAAJB,GAAsB,E;  
G;EAGzC,MAAM,QAAS,GAAE,iB;IACb,OAAO,MAAM,OAAO,CAAC,KA  
AK,GAAC,CAAP,C;G;EAGxB,MAAM,QAAS,GAAE,iB;IACb,OAAO,MAAM,OAAO,CAAC,KA  
AK,GAAC,CAAP,C;G;EAGxB,MAAM,KAAM,GAAE,IAAI,KAAM,IAAG,I;EAE3B,MAAM,aAAc,GAAE,I;EAEtB,oB;IACI,OAAyB,CAAhB,CAAE,GAAE,YAA  
Y,KAAG,CAAE,GAAE,KAAP,CAAE,GAAe,CAAZ,CAAE,GAAE,KAAQ,KAAG,CAAE,GAAE,CAAP,CAAW,  
GAAE,C;G;EA6DtE,CA1DD,Y;IACG,IAAI,MAAM,IAAI,WAAJ,CAAgB,CAAhB,C;IACV,IAAI,aAAa,IAAI,YA  
AJ,CAAiB,GAAjB,C;IACjB,IAAI,aAAa,IAAI,YAAJ,CAAiB,GAAjB,C;IACjB,IAAI,WAAW,IAAI,UAAJ,CAAE,  
GAAf,C;IACf,IAAI,WAAW,C;IACf,IAAI,YAA  
Y,C;IAEhB,UAAU,CAAC,CAAD,CAAI,GAAE,E;IACb,IAAI,QAAQ,CAAC,QAAD,CAAW,KA  
AI,CAA3B,C;MACI,QAAS,GAAE,C;MACX,SAAU,GAAE,C;;IAGhB,MAAM,aAAc,GAAE,iB;MACIB,OAAO,MAAM,gBAAgB,CAAC,KA  
AK,CAAC,KAAD,CAAQ,GAAE,GA  
AF,GAAQ,KAAtB,C;K;IAGjC,MAAM,gBAAiB,GAAE,iB;MACrB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHB,OAAO,MAAM,KA  
AK,SAAS,CAAC,QA  
AQ,CAAC,QAAD,CAAT,EAAqB,QA  
AQ,CAAC,SAAD,CAA7B,C;K;IAG/B,MAAM,eAAgB,GAAE,iB;MACpB,QA  
AQ,CAAC,QAAD,CAAW,GAAE,KA  
AK,K;MAC1B,QA  
AQ,CAAC,SAAD,CAAY,GAAE,KA  
AK,M;MAC3B,OAAO,UAAU,CAAC,CAAD,C;K;IAGrB,MAAM,YAAa,GAAE,iB;MACjB,OAAO,MAAM,eAAe,CAAC,KA  
AK,CAAC,KAAD,CAAQ,GAAE,GA  
AF,GAAQ,KAAtB,C;K;IAGhC,MAAM,eAAgB,GAAE,iB;MACpB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHB,OAAO,QA  
AQ,CAAC,CAAD,C;K;IAGnB,MAAM,cAAe,GAAE,iB;MACnB,QA  
AQ,CAAC,CAAD,CAAI,GAAE,K;MACd,OAAO,UAAU,CAAC,CAAD,C;K;IAIrB,MAAM,cAAe,GAAE,iB;MACnB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHB,OAAO,QA  
AQ,CAAC,SAAD,CAAY,GAAE,a;K;IAGjC,MAAM,eAAgB,GAAE,e;MACpB,IAAc,CAAT,GA  
AI,GAAE,CAAG,MAAI,GA  
AI,B,C;QACI,OAAO,GA  
AI,GAAE,C;;QAGb,UAAU,CAAC,CAAD,CAAI,GAAE,G;QACHB,OAAc,CAA9B,QA  
AQ,CAAC,SAAD,CAAY,GAAE,EAAG,GAAE,CAAG,IAAE,QA  
AQ,CAAC,QAAD,CAAW,GAAE,C;;K;GAGvE,G;EAEF,MAAM,cAAe,GAAE,a;IACnB,OAAO,CAAE,IAAG,IAAK,GAAE,CAAF,GAAM,MAAM,SAAS,E;G;E  
C7G1C,IAAI,OAAO,MAAM,UAAU,WAA  
Y,KA  
AI,WAA3C,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAU,EA  
AmB,YAA  
nB,EA  
AiC,QAC3C,kC;MACH,QAAS,GAAE,QAAS,IAAG,C;MACvB,OAAO,IAAI,YAA  
Y,CAAC,YA  
AD,EA  
Ae,QA  
Af,CAAyB,KA  
AI,Q;KAHN,CAAjC,C;;EAOzB,IAAI,OAAO,MAAM,UAAU,SAAU,KA  
AI,WAAzC,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAU,EA  
AmB,UAA  
nB,EA  
A+B,QACzC,kC;MACH,IAAI,gBAAgB,IAAI,SAAS,E;MACjC,IAAI,QAAS,KA  
AI,SAAU,IAAG,QAAS,GAAE,aAAa,OAAtD,C;QACI,QAAS,GAAE,aAAa,

O;;MAE5B,QAAS,IAAG,YAAY,O;MACxB,IAAI,YAAY,aAAa,QAAQ,CAAC,YAAD,EAAe,QAAf,C;MACrC,O  
AAO,SAAU,KAAI,EAAG,IAAG,SAAU,KAAI,Q;KARG,CAA/B,C;;EAazB,IAAI,OAAO,IAAI,KAAM,KAAI,W  
AAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,CAAE,GAAE,CAAC,C;MACL,IAAI,CAAE,KAAI,CAAE,IAAG,K  
AAK,CAAC,CAAD,CAApB,C;QACI,OAAO,MAAM,CAAC,CAAD,C;;MAEjB,OAAO,CAAE,GAAE,CAAE,GA  
AE,CAAF,GAAM,E;K;;EAG3B,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,a;MACT  
,IAAI,KAAK,CAAC,CAAD,CAAT,C;QACI,OAAO,G;;MAEX,IAAI,CAAE,GAAE,CAAR,C;QACI,OAAO,IAAI,  
MAAM,CAAC,CAAD,C;;MAErB,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAuKtB,CAnKD,Y;IACG,IAAI,UA  
AU,qB;IACd,IAAI,iBAaiB,IAAI,KAAK,CAAC,OAAD,C;IAC9B,IAAI,iBAaiB,IAAI,KAAK,CAAC,cAAD,C;IA  
C9B,IAAI,uBAAuB,CAAC,GAAC,c;IAC7B,IAAI,uBAAuB,CAAC,GAAC,c;IAE7B,IAAI,OAAO,IAAI,KAAM,K  
AAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UA  
CI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAA  
E,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAEP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAD,C;UACb,IA  
AI,KAAK,CAAE,GAAE,C;UACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAb,C;YAAkB,OAAO,IAAI,IAAI,CAA  
C,CAAE,GAAE,IAAI,IAAT,C;UACjC,IAAI,CAAC,QAAQ,CAAC,EAAD,CAAAb,C;YAAmB,OAAO,CAAC,IAAI  
,IAAI,CAAC,CAAC,CAAE,GAAE,IAAI,IAAV,C;UACnC,OAAGB,CAAR,CAAE,GAAE,EAAl,IAAE,C;;O;;IAI9  
B,IAAI,OAAO,IAAI,KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,IAAI,CAA  
C,CAAD,C;QACb,IAAI,KAAK,CAAE,GAAE,C;QACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAC  
,QAAQ,CAAC,EAAD,CAA7B,C;UAAmC,OAAO,IAAI,IAAI,CAAC,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IA  
AI,IAAnB,C;QACID,OAAGB,CAAR,CAAE,GAAE,EAAl,IAAE,C;O;;IAI1B,IAAI,OAAO,IAAI,KAAM,KAAI,W  
AAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI  
,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAA  
E,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAGP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAC,CAAF,CAAhB,EAAs  
B,IAAI,IAAI,IAAI,CAAC,CAAC,CAAF,C;UACIC,OAAO,CAAE,KAAI,QAAS,GAAE,CAAF,GAAM,CAAE,KA  
AI,QAAS,GAAE,EAfF,GAAe,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAAE,CAAP,C;;O;;IAQeE,IAAI,OA  
AO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,QAAQ,a;QACR,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAEI,IAAI  
,CAAE,GAAE,oBAAR,C;YAEI,IAAI,CAAE,GAAE,oBAAR,C;cAGI,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GA  
AE,IAAI,I;;cAKzB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAE,GAAG,CAAE,IAAG,CAAE,GAAE,CAAP,C  
AAZ,C;;YAKnB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CA  
AT,CAAd,C;;eAGIB,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAED,OAAO,CAAC,KAAK,CAAC,CAAC,CAAF,C;;  
UAKb,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,IAAG,cAAAnB,C;YAEI,IAAI,KAAK,CAAE,G  
AAE,CAAE,GAAE,C;YAEjB,MAAO,IAAG,EAAG,GAAE,C;;UAEEnB,OAAO,M;;O;MAGf,IAAI,MAAO,GAAE,  
K;;IAEjB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,CAAE,GAAE,  
CAAR,C;UAEI,OAAO,G;eAEN,IAAI,CAAE,GAAE,CAAE,IAAG,cAAAb,C;UAED,IAAI,CAAE,GAAE,oBAAR,C  
;YAGI,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,I;;YAlzB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,I  
AAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CAAT,CAAd,C;;UAKnB,IAAI,IAAI,IAAI,KAAK,CAAC,CAA  
E,GAAE,CAAL,C;UAEjB,IAAI,SAAS,C;UACb,IAAI,CAAE,IAAG,cAAT,C;YAEI,IAAI,KAAK,CAAE,GAAE,C  
AAE,GAAE,C;YAEjB,MAAO,IAAG,EAAG,GAAE,E;;UAGnB,OAAO,IAAI,KAAK,CAAC,CAAD,CAAI,GAAE  
,M;;O;;IAIIC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,  
CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,  
cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;QAEX,OAAO,I  
AAI,IAAI,CAAS,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAAE,CAAP,CAAT,CAAoB,GAAE,C;O;;IAG7C,I  
AAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,  
CAAI,GAAE,cAAIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,K  
AAK,EAAG,GAAE,C;UAEd,OAAG,CAAC,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,G  
AAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAL,C;O;;IAGvB,IAAI,OAAO,IAAI,  
MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cA  
AIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GA  
AE,C;UAEd,OAAG,EAAG,GAAE,EAAG,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;Q

AExC,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,C;O;;GAG/B,G;EACF,IAAI,OAAO,IAAI,MAAO,KAAI,W  
AA1B,C;IACI,IAAI,MAAO,GAAE,Y;MACT,IAAI,IAAI,C;MACR,IAAI,SAAS,SAAS,O;MAEtB,KAAK,IAAI,IA  
AI,CAAb,EAAGB,CAAE,GAAE,MAApB,EAA4B,CAAC,EAA7B,C;QACI,IAAI,SAAS,CAAC,CAAD,CAAI,KA  
AI,QAAS,IAAG,SAAS,CAAC,CAAD,CAAI,KAAI,CAAC,QAAnD,C;UACI,OAAO,Q;;QAEX,CAAE,IAAG,SA  
AS,CAAC,CAAD,CAAI,GAAE,SAAS,CAAC,CAAD,C;;MAEjC,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAGx  
B,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,a;MACT,OAAO,IAAI,IAAI,CAAC,CA  
AD,CAAI,GAAE,IAAI,O;K;;EAGjC,IAAI,OAAO,IAAI,KAAM,KAAI,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;M  
ACR,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,M;K;;EAGjC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA  
1B,C;IACI,IAAI,MAAO,GAAG,oB;MACV,OAAO,a;QACH,IAAI,SAAS,CAAE,KAAI,C;QACnB,IAAI,MAAO,  
KAAI,CAAf,C;UACI,OAAO,E;;QAEX,OAAO,EAAG,IAAG,GAAG,CAAC,MAAD,CAAS,GAAE,GAAI,GAAE,  
CAAvB,CAA0B,GAAE,C;O;KAE5C,CAAC,IAAI,IAAL,EAAW,IAAI,IAAf,C;;EAIN,IAAI,OAAO,WAAW,OAA  
Q,KAAI,WAAIC,C;IACI,WAAW,OAAQ,GAAE,a;MACjB,OAAO,CAAE,IAAG,IAAK,IAAG,CAAC,UAAW,IA  
AG,IAAK,IAAG,CAAC,UAAU,UAAW,KAAI,SAAS,UAAU,U;K;;EAIhG,IAAI,OAAO,KAAK,UAAU,KAAM,K  
AAI,WAApC,C;IAEI,MAAM,eAAe,CAAC,KAAK,UAAW,EAAB,MAAIB,EAA0B,QACpC,iB;MAGH,IAAI,IA  
AK,IAAG,IAAZ,C;QACI,MAAM,IAAI,SAAJ,CAAc,6BAAd,C;;MAGV,IAAI,IAAI,MAAM,CAAC,IAAD,C;MA  
Gd,IAAI,MAAM,CAAC,OAAQ,KAAI,C;MAGvB,IAAI,QAAQ,SAAS,CAAC,CAAD,C;MACrB,IAAI,gBAAGB,  
KAAM,IAAG,C;MAG7B,IAAI,IAAI,aAAc,GAAE,CAAE,GACIB,IAAI,IAAI,CAAC,GAAI,GAAE,aAAP,EAA5B  
,CAAtB,CADU,GAEIB,IAAI,IAAI,CAAC,aAAD,EAAGB,GAAhB,C;MAGhB,IAAI,MAAM,SAAS,CAAC,CAAD  
,C;MACnB,IAAI,cAAc,GAAI,KAAI,SAAU,GACIB,GADkB,GACZ,GAAI,IAAG,C;MAG/B,IAAI,aAAa,WAAy,  
GAAE,CAAE,GACHB,IAAI,IAAI,CAAC,GAAI,GAAE,WAAP,EAAoB,CAApB,CADQ,GAeHB,IAAI,IAAI,CAA  
C,WAAD,EAAC,GAAd,C;MAGzB,OAAO,CAAE,GAAE,UAAx,C;QACI,CAAC,CAAC,CAAD,CAAI,GAAE,K;  
QACP,CAAC,E;;MAIL,OAAO,C;KAvCgC,CAA1B,C;;EA4HvB,CahFD,Y;IACG,yC;MACI,IAAI,MAAO,GAAE  
,CAAb,C;QAAGB,OAAO,IAAI,IAAI,CAAC,CAAD,EAAI,MAAO,GAAE,MAAb,C;MAC/B,OAAO,IAAI,IAAI,C  
AAC,MAAD,EAAS,MAAT,C;K;IAEnB,qC;MACI,IAAI,OAAO,GAAI,KAAI,WAAAnB,C;QACI,GAAI,GAAE,IA  
AI,O;;MAEd,KAAM,GAAE,eAAe,CAAC,KAAM,IAAG,CAAV,EAAa,IAAI,OAAjB,C;MACvB,GAAI,GAAE,IA  
AI,IAAI,CAAC,KAAD,EAAQ,eAAe,CAAC,GAAD,EAAM,IAAI,OAAV,CAAvB,C;MACd,OAAO,IAAI,IAAI,Y  
AAR,CAAqB,IAAI,SAAS,CAAC,KAAD,EAAQ,GAAR,CAAI,C;K;IAGX,IAAI,SAAS,CAAC,SAAD,EAAY,U  
AAZ,EAAwB,WAAxB,EAAqC,UAArC,EAAdD,YAAjD,EAA+D,YAA/D,C;IACb,KAAK,IAAI,IAAI,CAAb,EAA  
gB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACv  
B,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAAuB,MA  
AvB,EAA+B,QACzC,KAAK,UAAU,KAD0B,CAA/B,C;;MAIzB,IAAI,OAAO,UAAU,UAAU,MAAO,KAAI,WA  
A1C,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAAuB,OAAvB,EAAGC,QAC1C,eAD0C,CAAhC,C;;;MAQJ,  
CAApB,Y;OAAc,MAAM,CAAC,IAAD,EAAO,IAAI,UAAJ,CAAE,CAAf,CAAP,E;;MAErB,IAAI,QAAQ,QAAQ,  
UAAU,M;MAC9B,MAAM,eAAe,CAAC,QAAQ,UAAT,EAAqB,OAArB,EAA8B,QACxC,uB;QACH,OAAO,KA  
AK,KAAK,CAAC,IAAD,EAAO,IAAP,EAAa,EAAE,MAAM,KAAK,CAAC,KAAD,CAA1B,C;OAF0B,CAA9B,C  
;;IASzB,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI  
,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,IAAK,KAAI,WAAxC,C;QACI,MAAM,eA  
Ae,CAAC,UAAU,UAAx,EAAuB,KAAvB,EAA8B,QACxC,0B;UACH,OAAO,EAAE,MAAM,KAAK,CAAC,IAA  
D,CAAM,IAAI,CAAC,QAAD,EAAW,IAAX,C;SAFa,CAA9B,C;;IAU7B,IAAI,uBAAuB,gB;MACvB,IAAI,CAA  
E,GAAE,CAAR,C;QAAW,OAAO,E;MACIB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,C;MAEIB,IAAI,CAA  
E,KAAI,CAAV,C;QACI,IAAI,CAAE,KAAI,CAAV,C;UAAa,OAAO,C;QAEpB,IAAI,KAAK,CAAE,GAAE,C;QA  
Cb,OAAO,EAAG,KAAI,CAAE,GAAE,CAAE,GAAE,CAAF,GAAG,EAAG,GAAE,CAAE,GAAE,EAFF,GAAG,  
C;;MAG7C,OAAO,CAAE,KAAI,CAAE,GAAG,CAAE,KAAI,CAAE,GAAE,CAAF,GAAM,CAAjB,GAASB,E;K;  
IAGzC,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,  
aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eA  
Ae,CAAC,UAAU,UAAx,EAAuB,MAAvB,EAA+B,QACzC,2B;UACH,OAAO,KAAK,UAAU,KAAK,KAAK,CA  
AC,IAAD,EAAO,eAAgB,IAAG,oBAA1B,C;SAFY,CAA/B,C;;GAO/B,G;ECxXF,MAAM,KAAM,GAAE,QACH,  
OADG,aAEC,WAFD,UAGF,QAHE,C;EAMd,MAAM,WAAy,GAAE,2C;IACb,IAAI,qBAAGB,MAAM,yBAAY

B,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;  
MACI,OAAO,kBAAkB,IAAI,KAAC,CAAC,UAAD,C;;IAGtC,kBAAmB,GAAE,MAAM,yBAAYB,CAAC,UAAD  
,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,OAAO,UAAU,CA  
AC,YAAD,C;;IAGrB,OAAO,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAaIC,EAA2  
C,YAA3C,C;G;EAG5B,MAAM,WAAy,GAAE,kD;IACHb,IAAI,qBAAqB,MAAM,yBAAYB,CAAC,KAAD,EAA  
Q,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;MACI,kBAAkB,IAAI,K  
AAK,CAAC,UAAD,EAAa,KAAb,C;MAC3B,M;;IAGJ,kBAAmB,GAAE,MAAM,yBAAYB,CAAC,UAAD,EAAa,  
YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,UAAU,CAAC,YAAD,CA  
Ae,GAAE,K;MAC3B,M;;IAGJ,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAaIC,EAA  
2C,YAA3C,EAAyD,KAAzD,C;G;EAGrB,iD;IACI,IAAI,IAAK,KAAL,KAAb,C;MAAOB,OAAO,I;IAE3B,IAAI,W  
AAW,IAAI,W;IACnB,IAAI,QAAS,IAAG,IAAhB,C;MACI,IAAI,aAAa,QAAQ,W;MACzB,KAAK,IAAI,IAAI,CA  
Ab,EAAgB,CAAE,GAAE,UAAU,OAA9B,EAAuC,CAAC,EAAxC,C;QACI,IAAI,0BAA0B,CAAC,UAAU,CAAC  
,CAAD,CAAX,EAAgB,KAAhB,CAA9B,C;UACI,OAAO,I;;;IAKnB,IAAI,iBAAiB,IAAI,UAAW,IAAG,IAAK,G  
AAE,MAAM,eAAe,CAAC,IAAI,UAAU,CAAyB,GAA0C,I;IACtF,IAAI,mBAAmB,cAAe,IAAG,IAAK,GAAE,cA  
Ac,YAAhB,GAA+B,I;IAC7E,OAAO,gBAAiB,IAAG,IAAK,IAAG,0BAA0B,CAAC,gBAAD,EAAMB,KAAhB,C;  
G;EASjE,MAAM,OAAQ,GAAE,yB;IACZ,IAAI,KAAM,KAAL,MAAd,C;MACI,QAAQ,OAAO,MAAf,C;aACS,Q;  
aACA,Q;aACA,S;aACA,U;UACD,OAAO,I;;UAEP,OAAO,MAAO,YAAW,M;;IAIrC,IAAI,MAAO,IAAG,IAAK,  
IAAG,KAAM,IAAG,IAAK,KAAL,OAAO,MAAO,KAAL,QAAS,IAAG,OAAO,MAAO,KAAL,UAApD,CAApC,C;  
MACI,OAAO,K;;IAGX,IAAI,OAAO,KAAM,KAAL,UAAW,IAAG,MAAO,YAAW,KAArD,C;MACI,OAAO,I;;IA  
GX,IAAI,QAAQ,MAAM,eAAe,CAAC,KAAD,C;IACjC,IAAI,cAAc,KAAM,IAAG,IAAK,GAAE,KAAK,YAAP,  
GAAhB,I;IACtD,IAAI,WAAy,IAAG,IAAK,IAAG,YAAa,IAAG,WAA3C,C;MACI,IAAI,WAAW,WAAW,W;MA  
C1B,IAAI,QAAQ,KAAM,KAAL,MAAM,KAAK,OAAjC,C;QACI,OAAO,MAAO,KAAL,K;;IAI1B,IAAI,gBAAg  
B,KAAK,W;IAGzB,IAAI,aAAc,IAAG,IAArB,C;MACI,OAAO,MAAO,YAAW,K;;IAG7B,IAAI,aAAa,KAAM,K  
AAI,MAAM,KAAK,UAAW,IAAG,MAAM,YAAa,IAAG,IAA1E,C;MACI,OAAO,0BAA0B,CAAC,MAAM,YAA  
P,EAAqB,KAArB,C;;IAGrC,OAAO,K;G;EAGX,MAAM,SAAU,GAAE,a;IACd,OAAO,OAAO,CAAE,IAAG,QA  
AS,IAAG,CAAE,YAAW,MAAM,K;G;EAGtD,MAAM,OAAQ,GAAE,iB;IACZ,OAAO,KAAM,YAAW,MAAM,  
U;G;EAGiC,MAAM,aAAc,GAAE,iB;IACiB,IAAI,OAAO,OAAO,K;IAEiB,OAAO,IAAK,KAAL,QAAS,IACiB,IA  
AK,KAAL,SAAU,IACnB,MAAM,SAAS,CAAC,KAAD,CAAQ,IACvB,MAAM,OAAO,CAAC,KAAD,EAAQ,MA  
AM,OAAO,WAArB,C;G;EAGxB,MAAM,eAAgB,GAAE,iB;IACpB,OAAO,OAAO,KAAM,KAAL,QAAS,IAAG,  
MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM,OAAO,aAArB,C;G;;aCnDV,gB;;;ICrE3C,gB;MAkBI,4B;MA  
jBA,aAA6C,E;MAC7C,gBAAgD,C;K;4EAG5C,Y;MAAQ,iB;K;+EAGR,Y;MAAQ,oB;K;qCAEZ,iB;MAAyC,OA  
AQ,0BAAR,YAAQ,EAAU,KAAM,QAAhB,C;K;4BAEjD,iB;MAAmC,gBAAS,K;K;8BAE5C,Y;MAA+B,OAAhC  
,MAAmC,kBAA8B,IAA9B,C;K;8BAE/B,Y;MAA0B,gB;K;IAE1B,0B;MAAA,8B;K;;IAAA,sC;MAAA,qC;QAA  
A,oB;;MAAA,8B;K;;IDfJ,mC;MAC4C,oBAAa,MAAS,IAAT,CAAb,EAA6B,SAA7B,C;K;gEAE5C,yB;MAAA,m  
B;MAAA,6B;QAC2D,YAAa,QAAS,IAAT,C;QAIvD,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAA  
U,CAAV,iB;UACI,MAAM,CAAN,IALgF,IAKrE,CAAK,CAAL,C;;QALwC,OAOhD,K;O;KARX,C;gEAGA,uB;  
MAEiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,  
KAAK,CAAL,C;;MAEf,OAAO,K;K;IAGX,kC;MAIiB,IAAN,I;MAFP,aAAsB,MAAe,IAAf,C;MACtB,gBAAkB,c;  
MAEd,IADS,IACt,mBADs,IACt,EAAM,IAAN,E;QAAC,oBAAa,MAAb,EAAqB,KAArB,C;WACd,WAFS,IAET  
,S;QAAS,a;;QAZA,U;QAAA,SAaqB,Mabf,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UAakC,MAZ9B,CA  
AM,CAAN,IAysC,IAZ3B,CAAK,CAAL,C;;QAYH,OAAhB,M;;MAHiC,W;K;2EAOJ,yB;MAAA,iC;MAAA,6B;  
QACoF,YAAa,aAAa,IAAb,EAAMB,KAAhB,C;QAIbHf,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,a  
AAU,CAAV,iB;UACI,MAAM,CAAN,IAiBoH,IAjBzG,CAAK,CAAL,C;;QAIbIE,OafzE,K;O;KAcX,C;IAGA,+B  
;MAKiB,IAAN,I;MAFP,aAAa,IAAb,WAAa,CAAD,IAAC,C;MACb,gBAAkB,W;MAEd,IADS,IACt,mBADs,IA  
CT,EAAM,IAAN,YADS,IACt,EAAY,KAAZ,E;QAAqB,a;;QAIbZ,U;QAAA,SA2BkB,MA3BZ,OAAN,GAAa,C  
AAb,I;QAAb,aAAU,CAAV,mB;UA2B+B,MA1B3B,CAAM,CAAN,IA0BmC,IA1BxB,CAAK,CAAL,C;;QA0BH,  
OAAmB,M;;MAF/B,W;K;qEAMJ,yB;MAAA,2B;MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAy,UAAU,IAAV,  
EAAgB,IAAhB,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CA

AL,E;UACpB,KAAK,CAAC,CAAD,CAAG,GAAG,K;;QAEP,OAAO,K;O;KARX,C;mFAWA,yB;MAAA,mB;M  
AAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAy,QAAY,IAAZ,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QA  
Ab,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;UACpB,KAAK,CAAC,CAAD,CAAG,GAAG,K;;QAEP,OAA  
O,K;O;KARX,C;IAWA,+B;MAIiB,IAAN,I;MAFP,aAAsB,MAAY,IAAZ,C;MACTb,gBAaKb,W;MAEd,IADS,IA  
CT,mBADs,IACT,EAAM,IAAN,E;QAAC,oBAAa,MAAb,K;WACd,WAFS,IAET,S;QAAS,a;;QA3DA,U;QAAA,S  
A4DKB,MA5DZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UA4D+B,MA3D3B,CAAM,CAAN,IA2DmC,I  
A3DxB,CAAK,CAAL,C;;QA2DH,OAAMb,M;;MAH/B,W;K;qEAOJ,yB;MAAA,2B;MAAA,6B;QAC2E,YAAa,U  
AAU,IAAV,EAAGB,KAAhB,C;QAJEvE,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB  
;UACI,MAAM,CAAN,IAgEwG,IAhE7F,CAAK,CAAL,C;;QAgEwD,OA9DhE,K;O;KA6DX,C;IAGA,wC;MACiB  
,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,S;;MAEf  
,OAAO,K;K;IEIFX,iC;MAAA,qC;MAEI,iBAC8B,Q;MAE9B,iBAC8B,sB;MAE9B,yBAEsC,MAAM,G;MAE5C,y  
BAEsC,CAAC,GAAD,GAAO,G;MAE7C,WAEwB,EAAE,MAAM,GAAR,C;MAExB,kBACuB,C;MAEvB,iBACs  
B,E;K;;IAxB1B,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IA2BA,gC;MAAA,oC;MAEI,iBAC6B,O;MAE7B,iBAC  
6B,Y;MAE7B,yBAEqC,MAAO,G;MAE5C,yBAEqC,CAAC,GAAD,GAAQ,G;MAE7C,WAEuB,EAAE,MAAO,G  
AAT,C;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;IAxB1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IA2BA,8B;  
MAAA,kC;MAEI,iBACqB,W;MAErB,iBACqB,U;MAErB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,0C;MAAA,  
yC;QAAA,wB;;MAAA,kC;K;IAeA,+B;MAAA,mC;MAEI,iBACJ,MAAM,KAAoB,U;MAEtB,iBACJ,MAAM,KA  
AoB,U;MAEtB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,gC;M  
AAA,oC;MAEI,iBACuB,U;MAEvB,iBACuB,K;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,4C;MAAA,2  
C;QAAA,0B;;MAAA,oC;K;IAeA,+B;MAAA,mC;MAEI,iBACsB,Q;MAEtB,iBACsB,G;MAEtB,kBACuB,C;MAE  
vB,iBACsB,C;K;;IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,+B;MAAA,mC;MAEI,iBACmC,C;MAE  
nC,iBACmC,K;MAEnC,0BAC4C,K;MAE5C,0BAC4C,K;MAE5C,yBAC2C,K;MAE3C,yBAC2C,K;MAE3C,qBA  
CuC,uB;MAEvC,qBACuC,sB;MAEvC,kBACuB,C;MAEvB,iBACsB,E;K;;IA9B1B,2C;MAAA,0C;QAAA,yB;;M  
AAA,mC;K;IAiCA,iC;MAAA,qC;K;;IAAA,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAEA,kC;MAAA,sC;K;;IA  
AA,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;aCkkuBoB,gB;;cC/ntB0C,mB;;gBAyEvC,  
yB;eAAyB,wB;;uBAgBzB,gC;sBAAwB,+B;mCA4JjC,qB;mCA5ImC,qB;kBAQ1B,2B;iBAA0B,0B;;eC3YgB,  
wB;sBCoBA,sB;iBCnBA,0B;;aC5P8B,e;;gCCIDhD,yC;+BCVA,uC;+BCAA,sC;;gCCyJ/B,+B;+BAIW,sC  
;gCCqWc,+B;0BAHvB,kC;uBAr6BO,gC;yBA8WD,iC;0BACA,mC;yBA4JA,iC;gCAmZP,oC;+BAbc,oC;+BAEC  
,+B;yBAEQ,kC;;gBCr0C6C,yB;;  
;IC/ErF,kD;MAMuF,wC;K;IANvF,4CAOI,  
Y;MAAuC,8B;K;IAP3C,8E;ICGA,kD;MAQuF,wC;K;IARvF,4CASI,Y;MAAuC,8B;K;IAT3C,8E;0FbOA,qB;MA  
QI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,  
C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,  
OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;  
K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,O  
AAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;  
4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OA  
AO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0  
FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAA  
O,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4F  
AGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,  
UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAG  
X,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UA  
AI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,  
qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAA  
I,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB  
;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,C  
AAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;M

AQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;IAGX,sC;MAII,OAAO,mBAAQ,OAA  
R,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,K  
AAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAA  
oB,C;K;IAG/B,wC;MAOI,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAOI,OAAO,qBAAQ,OAAR,KAAo  
B,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C  
;K;oGAK/E,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA  
AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IA  
Ac,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MA  
AA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;  
O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAA  
I,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CA  
AT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,  
8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,K  
AAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc  
C,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SA  
AS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;  
MAAA,8D;MAAA,gC;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAA  
tC,GAAcD,uBAAa,KAAb,E;O;KAPjE,C;oGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,  
KAAV,C;O;KAPhB,C;qGAUA,yB;MAAA,qD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;KAP  
hB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;  
MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAA  
A,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA  
W,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,E  
AAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;  
KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAW,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;8EAUA  
,gC;MAOW,sB;;QAyB,S,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAzbH,SAybO,CAAU,OAAV,CAAJ,C;YAAwB  
,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA1bP,yB;K;gFAGJ,gC;MAOW,sB;;QAubS,Q;QAaHb,iD;UAAgB,  
cAAhB,e;UAAsB,IAvbH,SAubO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;M  
AxbP,yB;K;gFAGJ,gC;MAOW,sB;;QAqbS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IArbH,SAqbO,CAAU,OAA  
V,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MATbP,yB;K;gFAGJ,gC;MAOW,sB;;QAmbS,Q;  
QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IANbH,SambO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;Q  
AC9C,qBAAO,I;;;MApbP,yB;K;gFAGJ,gC;MAOW,sB;;QAibS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAjbH,S  
AibO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA1bP,yB;K;gFAGJ,gC;MAO  
W,sB;;QA+aS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA+aH,SA+aO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,  
O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAhbP,yB;K;gFAGJ,gC;MAOW,sB;;QA6aS,Q;QAaHb,iD;UAAgB,cAAhB,e;  
UAAsB,IA7aH,SA6aO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA9aP,yB;K  
;gFAGJ,gC;MAOW,sB;;QA2aS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA3aH,SA2aO,CAAU,OAAV,CAAJ,C;  
YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA5aP,yB;K;gFAGJ,yB;MA4aA,oC;MAAA,gC;MA5aA,uC;  
QAOW,sB;;UAyaS,Q;UAaHb,iD;YAAgB,cAAhB,0B;YAAsB,IAzaH,SAyaO,CAAU,oBAAV,CAAJ,C;cAAwB,q  
BAAO,O;cAAP,uB;;;UAC9C,qBAAO,I;;;QA1aP,yB;O;KAPJ,C;sFAUA,yB;MAw1CA,0D;MAAA,+C;MAx1CA,u  
C;QAOW,qB;;UAu1CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc  
,UAAK,KAAL,C;YACd,IAz1Cc,SAy1CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAA  
O,I;;;QA31CP,wB;O;KAPJ,C;wFAUA,yB;MA21CA,0D;MAAA,+C;MA31CA,uC;QAOW,qB;;UA01CO,Q;UAAA  
,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA51Cc,  
SA41CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA91CP,wB;O;KAPJ,C;wF  
AUA,yB;MA81CA,0D;MAAA,+C;MA91CA,uC;QAOW,qB;;UA61CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;  
UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA1CC,SA+1CV,CAAU,OAAV,CAAJ,C;  
cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAj2CP,wB;O;KAPJ,C;wFAUA,yB;MAi2CA,0D;MAAA,+C;

MAj2CA,uC;QAOW,qB;;UAg2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAI2Cc,Sak2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAp2CP,wB;O;KAPJ,C;wFAUA,yB;MAo2CA,0D;MAAA,+C;Map2CA,uC;QAOW,qB;;UAm2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAr2Cc,SAq2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAv2CP,wB;O;KAPJ,C;wFAUA,yB;MAu2CA,0D;MAAA,+C;MAv2CA,uC;QAOW,qB;;UAs2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAx2Cc,SAw2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA12CP,wB;O;KAPJ,C;wFAUA,yB;MA02CA,0D;MAAA,+C;MA12CA,uC;QAOW,qB;;UAY2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA32Cc,SA22CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA72CP,wB;O;KAPJ,C;wFAUA,yB;MA62CA,0D;MAAA,+C;MA72CA,uC;QAOW,qB;;UA42CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA92Cc,SA82CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAh3CP,wB;O;KAPJ,C;wFAUA,yB;MAg3CA,0D;MAAA,+C;MAAA,oC;MAh3CA,uC;QAOW,qB;;UA+2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAj3Cc,SAi3CV,CAAU,oBAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAn3CP,wB;O;KAPJ,C;IAUA,0B;MAKI,IA4uNO,qBAAQ,CA5uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IA0uNO,qBAAQ,CA1uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAwuNO,qBAAQ,CAxuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IASuNO,qBAAQ,CAtuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAouNO,qBAAQ,CApuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAkuNO,qBAAQ,CAluNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAguNO,qBAAQ,CAhuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IA8tNO,qBAAQ,CA9tNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IA4tNO,qBAAQ,CA5tNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;kFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;kFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;kGASA,yB;MAAA,iE;MAAA,uC;QASW,Q;QAAA,+B;;UAYS,U;UAaHb,uD;YAAgB,cAaHb,iB;YACI,aAbwB,SAaX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIbA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,8DAAvB,C;;QAaHd,OAAO,I;O;KATX,C;8GAYA,gC;MASoB,Q;MAaHb,wBAAgB,SAaHb,gB;QAaG,cAAA,SAaHb,M;QACI,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,gC;MAIL,OAoiNO,qBAAQ,CapiNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAIL,OAqiNO,qBAAQ,CariNR,GAAe,IA





W,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;8FAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK  
,KAAL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,K  
AAL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KA  
AL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAA  
L,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,  
CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,C  
AAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CA  
AV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,yB;MAAA,oC;MAAA,uC;QAI,wD;UACI,IAAI,UAAU  
,sBAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,0D;MAAA,+  
C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,U  
AAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAA  
A,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IA  
AI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;  
MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UAC  
V,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,  
0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;U  
ACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MA  
AA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,u  
B;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;  
MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;U  
AAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FA  
YA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;  
UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FA  
YA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;  
UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FA  
YA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,  
C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8  
FAYA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QA  
Ad,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,sBAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAA  
O,E;O;KATX,C;IA YA,yB;MAQI,IAg7LO,qBAAQ,CAh7Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO  
,UAAK,wBAAL,C;K;IAGX,2B;MAQI,IA26LO,qBAAQ,CA36Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,  
OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA56LO,qBAAQ,CAt6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MA  
CV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAi6LO,qBAAQ,CAj6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;  
MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA45LO,qBAAQ,CA55Lf,C;QACI,MAAM,2BAAuB,iBAA  
vB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAu5LO,qBAAQ,CAv5Lf,C;QACI,MAAM,2BAAuB,i  
BAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAk5LO,qBAAQ,CAI5Lf,C;QACI,MAAM,2BA  
AuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA64LO,qBAAQ,CA74Lf,C;QACI,MAAM  
,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAw4LO,qBAAQ,CAx4Lf,C;QACI,M  
AAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;gFAGX,yB;MAAA,0D;MAAA,+C;MAAA,iE;M  
AAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UAC  
V,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mD  
AAvB,C;O;KAZV,C;gFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,Y  
AAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAA  
U,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;IFaEa,yB;MAAA,0D;  
MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,  
cAAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,  
MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;IFaEa,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;  
QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAA  
L,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;i

FAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,C  
AAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YA  
AwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;  
MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UA  
CV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,m  
DAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,  
YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UA  
AU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0  
D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAA  
c,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QA  
EnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,  
uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAA  
c,UAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB  
,C;O;KAZV,C;IAeA,yC;MAKsB,UAMA,M;MAPIB,IAAI,eAAJ,C;QACKB,OAAQ,WAAR,sBAAQ,CAAR,W;QA  
Ad,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAK,KAAL,SAAJ,C;YACI,OAAO,K;;;QAID,SAAQ,WAAR,sBAA  
Q,CAAR,W;QAAd,OAAc,gBAAd,C;UAAc,2B;UACV,IAAI,gBAAW,UAAK,OAAAL,CAAX,CAAJ,C;YACI,OAA  
O,O;;;MAInB,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,  
C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,  
Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAA  
L,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;  
MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;  
K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,  
gBAAW,UAAK,KAAL,CAAX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAA  
Q,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UA  
CI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,  
cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;  
MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UA  
AK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,C  
AAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,  
OAAO,E;K;IAGX,+B;MAMI,OA8jLO,qBAAQ,CA9jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;  
IAGpC,iC;MAMI,OA6jLO,qBAAQ,CA7jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;  
MAMI,OA4jLO,qBAAQ,CA5jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA  
2jLO,qBAAQ,CA3jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA0jLO,qBA  
AQ,CA1jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAyjLO,qBAAQ,CAzjL  
R,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAwjLO,qBAAQ,CAXjLR,GAAe,I  
AAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAujLO,qBAAQ,CAvjLR,GAAe,IAAf,GAAY  
B,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAsjLO,qBAAQ,CAtjLR,GAAe,IAAf,GAAYB,UAAK,m  
BAAO,CAAP,IAAL,C;K;4FAGpC,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL  
,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OA  
AV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QA  
MkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UA  
AK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;  
MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,  
cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAE  
nC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,S  
AAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OA  
AV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAM  
kB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK



gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,M  
AAM,gCAAYB,kCAAZB,C;;MAHIB,W;K;IAOJ,6B;MAliB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,  
MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;  
MAHIB,W;K;IAOJ,6B;MAliB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;  
aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;MAHIB,W;K;IAOJ,6B;MAliB  
,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAA  
L,C;UAAL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;MAHIB,W;K;IAOJ,6B;MAliB,IAAN,I;MAAA,QAAM,gBA  
AN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM  
,gCAAYB,kCAAZB,C;;MAHIB,W;K;IAOJ,6B;MAliB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAA  
M,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;MAHI  
B,W;K;IAOJ,6B;MAliB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,  
C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;MAHIB,W;K;IAOJ,6B;MAliB,IAAN  
,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UA  
AL,K;;UACQ,MAAM,gCAAYB,kCAAZB,C;;MAHIB,W;K;oFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8  
B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,S  
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,  
SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,  
6E;O;KafX,C;oFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QA  
CpB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YAC  
I,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,CAAC,K  
AAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;  
MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAg  
B,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,  
C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEI  
B,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aA  
AmB,I;QACnB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CA  
AJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI  
,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;M  
AAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAGB,SAAhB,g  
B;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,  
gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAv  
B,C;QAEIB,OAAO,iE;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;Q  
AXP,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OA  
AV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAG  
hB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MA  
AA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAsB,I;QACtB,YAAY,K;QACZ,wBAAGB,S  
AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8  
BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,  
mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,U  
AST,M;QAXP,aAAuB,I;QACvB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,U  
AAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ  
,I,;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4D;O;KafX,C;qFakBA  
,yB;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QAC  
pB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAA  
J,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,gDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I,;;QAGhB,IAAI,  
CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4E;O;KafX,C;IAkBA,iC;MAII,OAAW,q  
BAAQ,CAAZ,GAAe,UAAK,CAAL,CAAF,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,  
CAAL,CAAF,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAF,GAA4B,I;K;IA

GvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAA Q,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAA L,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC, mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CA AZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;gGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAy,K;MACZ ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAA W,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO, M;K;gGAGX,gC;MAMoB,Q;MAFhB,aAAoB,I;MACpB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QA AQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aA AqB,I;MACrB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CA AJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;Q AAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAmB,I;MACnB,YAAy,K;MACZ,wBAA gB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAA O,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGA GX,gC;MAMoB,Q;MAFhB,aAAoB,I;MACpB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M; QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;M AGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;M ACrB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UA CI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OA AAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAaB,I;MACtB,YAAy,K;MACZ,wBAAgB,SAAhB ,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB ,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;M AMoB,Q;MAFhB,aAAuB,I;MACvB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IA AI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IA AI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAMo B,Q;QAFhB,aAAoB,I;QACpB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI, IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;CAAW,OAAO,I;YACIB,SAAS,O;YACT,QAAQ,I;QAGhB,I AAI,CAAC,KAAL,C;UAAy,OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,4B;McVqGI,IAAI,Ed+qGI,KAAK,Cc /qGT,CAAJ,C;QACI,cd8qGc,sD;Qc7qGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md8qGV,OAAO,oBAAoB,gBAA V,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McnrGI,IAAI,Ed2rGI,KAAK,Cc3rGT,CAAJ,C;QA CI,cd0rGc,sD;QczrGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md0rGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,I AAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc/rGI,IAAI,EdusGI,KAAK,CcvsGT,CAAJ,C;QACI,cdssGc,sD;Qcrs Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdssGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd, CAAPB,C;K;IAGX,8B;Mc3sGI,IAAI,EdmtGI,KAAK,CcntGT,CAAJ,C;QACI,cdktGc,sD;QcjtGd,MAAM,gCAAy B,OAAQ,WAAjC,C;;MdktGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX, 8B;McvtGI,IAAI,Ed+tGI,KAAK,Cc/tGT,CAAJ,C;QACI,cd8tGc,sD;Qc7tGd,MAAM,gCAAyB,OAAQ,WAAjC,C;; Md8tGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McnuGI,IAAI,Ed 2uGI,KAAK,Cc3uGT,CAAJ,C;QACI,cd0uGc,sD;QczuGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md0uGV,OAAO, sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc/uGI,IAAI,EduvGI,KAAK,CcVv GT,CAAJ,C;QACI,cdsvGc,sD;QcrvGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdsvGV,OAAO,sBAAoB,gBAAV, mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc3vGI,IAAI,EdmwGI,KAAK,CcnwGT,CAAJ,C;Q ACI,cdkwGc,sD;QcJwGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdkwGV,OAAO,sBAAoB,gBAAV,mBAAO,CA AP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McVwGI,IAAI,Ed+wGI,KAAK,Cc/wGT,CAAJ,C;QACI,cd8wGc,s D;Qc7wGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md8wGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EA AC,CAAd,CAApB,C;K;IAGX,gC;McNxiGI,IAAI,Ed2xGI,KAAK,Cc3xGT,CAAJ,C;QACI,cd0xGc,sD;QczxGd,MA AM,gCAAyB,OAAQ,WAAjC,C;;Md0xGV,OAAO,gBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAh B,C;K;IAGX,kC;Mc/xGI,IAAI,EduyGI,KAAK,CcVvGT,CAAJ,C;QACI,cdsyGc,sD;QcryGd,MAAM,gCAAyB,OA

AQ,WAAjC,C;;MdsyGV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;  
Mc3yGI,IAAI,EdmzGI,KAAK,CcnzGT,CAAJ,C;QACI,cdkzGc,sD;QcJzGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;  
MdkzGV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;McvzGI,IAAI,Ed  
+zGI,KAAK,Cc/zGT,CAAJ,C;QACI,cd8zGc,sD;Qc7zGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md8zGV,OAAO,k  
BAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mcn0GI,IAAI,Ed20GI,KAAK,Cc30  
GT,CAAJ,C;QACI,cd00Gc,sD;QcZ0Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md00GV,OAAO,kBAAGB,gBAAV,  
mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mc/0GI,IAAI,Edu1GI,KAAK,Ccv1GT,CAAJ,C;QAC  
I,cds1Gc,sD;Qcr1Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mds1GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IA  
AU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mc31GI,IAAI,Edm2GI,KAAK,Ccn2GT,CAAJ,C;QACI,cdk2Gc,sD;Qcj2  
Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mdk2GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd  
,CAAhB,C;K;IAGX,kC;Mcv2GI,IAAI,Ed+2GI,KAAK,Cc/2GT,CAAJ,C;QACI,cd82Gc,sD;Qc72Gd,MAAM,gCA  
AYB,OAAQ,WAAjC,C;;Md82GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IA  
GX,kC;Mcn3GI,IAAI,Ed23GI,KAAK,Cc33GT,CAAJ,C;QACI,cd03Gc,sD;Qcz3Gd,MAAM,gCAAYB,OAAQ,WA  
AjC,C;;Md03GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;gGAGX,yB;MAAA  
,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UA  
AK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA  
,yB;MAAA,8D;MAAA,2C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,  
UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAX  
X,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,I  
AAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO  
,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/  
B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;Q  
AGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WA  
A+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,  
IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,  
wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,Q  
AAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;Q  
AMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAA  
O,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;  
MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C  
;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,oC;  
MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,  
KAAL,EA AV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;wFAcA,yB;  
MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SA  
Ab,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,  
WAAL,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QA  
Fb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IA  
AK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YACL,WAAW,I;;;Q  
AE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QAC  
X,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAA  
C,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,  
yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,  
SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IA  
AK,WAAL,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q  
;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,  
IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YACL,WAAW,I;;;  
QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QA  
CX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CA

AC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YA CL,WA AW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WA AW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAA A,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YA CL,WA AW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WA AW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YA CL,IAAK,WAAL,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAL,IAAJ,C;YA CL,WA AW,I;;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAQiB,Q;QAFb,eAA e,K;QACf,WA AW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,QAAJ,C;YACI,IAA K,WAAL,iBAAJ,C;eACJ,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YACD,IAAK,WAAL,iBAAJ,C;YA CL,WA AW,I;;; QAE nB,OAAO,I;O;KafX,C;kFAkBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAmgBA,Q;QA AhB,iD;U AAgB,cAAhB,e;UAAsB,IAngBU,SAmgBN,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAngB1D ,OAogBO,W;O;KA1gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAogBA,Q;QA AhB,iD;UAA gB,cAAhB,e;UAAsB,IApgBa,SAogBT,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAp gB1D,OAq gBO,W;O;KA3gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAqgBA,Q;QA AhB,iD;UAAgB,c AAhB,e;UAAsB,IArgBc,SAqgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QArgB1D,OAsgBO, W;O;KA5gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAsgBA,Q;QA AhB,iD;UAAgB,cAAhB, e;UAAsB,IAtgBY,SAsgBR,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAtgB1D,OAu gBO,W;O; KA7gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAu gBA,Q;QA AhB,iD;UAAgB,cAAhB,e;UA AsB,IAvgBa,SAvgBT,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAv gB1D,OAvgBO,W;O;KA9 gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAwgBA,Q;QA AhB,iD;UAAgB,cAAhB,e;UAAs B,IAxgBc,SAwgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAxgB1D,OAygBO,W;O;KA/gB X,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAYgBA,Q;QA AhB,iD;UAAgB,cAAhB,e;UAAsB,IA zgBe,SAygBX,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAzgB1D,OA0gBO,W;O;KAhhBX,C;o FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA0gBA,Q;QA AhB,iD;UAAgB,cAAhB,e;UAAsB,IA1gBg B,SA0gBZ,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QA1gB1D,OA2gBO,W;O;KAjhBX,C;oFA SA,yB;MAAA,+D;MA2gBA,oC;MAAA,gC;MA3gBA,uC;QAMW,kBAAS,gB;QA2gBA,Q;QA AhB,iD;UAAgB,c AAhB,0B;UAAsB,IA3gBa,SA2gBT,CAAU,oBAAV,CAAJ,C;YAAwB,WAAY,WAAL,oBAAJ,C;;QA3gB1D,OA4 gBO,W;O;KAlhBX,C;gGASA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAsgTV,gB;QADb,YAA Y,C;QA CZ,iD;UAAa,WAAb,e;UA16SI,IApGmC,SAoG/B,EAk6SkB,cAl6SIB,EAk6SkB,sBA16SIB,Wak6S2B,IA16S3B,C AAJ,C;YAA2C,sBAk6SZ,IA16SY,C;;QA pG/C,OAsGO,W;O;KA9GX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQ W,kBAAGB,gB;QAqgTV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAAb,e;UA95SI,IAvGsC,SAuG1C,EA85SkB,cA 95SIB,EA85SkB,sBA95SIB,WA85S2B,IA95S3B,CAAJ,C;YAA2C,sBA85SZ,IA95SY,C;;QAvG/C,OAYGO,W;O; KAjHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAogTV,gB;QADb,YAA Y,C;QACZ,iD;UA Aa,WAAb,e;UA15SI,IA1GuC,SA0GnC,EA05SkB,cA15SIB,EA05SkB,sBA15SIB,WA05S2B,IA15S3B,CAAJ,C;Y AA2C,sBA05SZ,IA15SY,C;;QA1G/C,OA4GO,W;O;KApHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBA AgB,gB;QAmgTV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAAb,e;UA5SI,IA7GqC,SA6GjC,EAs5SkB,cAt5SIB,E As5SkB,sBA5SIB,WAs5S2B,IA5S3B,CAAJ,C;YAA2C,sBA5SZ,IA5SY,C;;QA7G/C,OA+GO,W;O;KA vHX,C; kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAkgTV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAAb, e;UA15SI,IAhHsC,SAgHIC,EAk5SkB,cAl5SIB,EAk5SkB,sBA15SIB,Wak5S2B,IA15S3B,CAAJ,C;YAA2C,sBAk5 SZ,IA15SY,C;;QA hH/C,OAKHO,W;O;KA1HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAig TV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAAb,e;UA94SI,IANHuC,SAmHnC,EA84SkB,cA94SIB,EA84SkB,sBA 94SIB,WA84S2B,IA94S3B,CAAJ,C;YAA2C,sBA84SZ,IA94SY,C;;QAnH/C,OAqHO,W;O;KA7HX,C;kGAWA,y B;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAggTV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAAb,e;UA14SI,IA tHwC,SAsHpC,EA04SkB,cA14SIB,EA04SkB,sBA14SIB,WA04S2B,IA14S3B,CAAJ,C;YAA2C,sBA04SZ,IA14 SY,C;;QAtH/C,OA wHO,W;O;KAhIX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QA+/SV,gB;Q ADb,YAA Y,C;QACZ,iD;UAAa,WAAb,e;UA4SI,IAzHyC,SAyHrC,EAs4SkB,cAt4SIB,EAs4SkB,sBA4SIB,WAs 4S2B,IA4S3B,CAAJ,C;YAA2C,sBA4SZ,IA4SY,C;;QAzH/C,OA2HO,W;O;KANIX,C;kGAWA,yB;MAAA,+D; MA2HA,gC;MAo4SA,oC;MA//SA,uC;QAQW,kBAAGB,gB;QA8/SV,gB;QADb,YAA Y,C;QACZ,iD;UAAa,WAA







C;QAAe,OAAO,W;MACtB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAA K,WAAI,UAAI,KA AJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAA wB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MACG,yB;M AAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KA AJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB ,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACt B,WAAW,iBAAkB,IAAlB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KA AJ,CA AJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IA AI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAmB,IAAnB,C;MACG,yB;MAAd,OAAc,cAAd,C;QA Ac,uB;QACV,IAAK,WAAI,UAAI,KA AJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wB AAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAgB,IAA hB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,KA AJ,EAAJ,C;;MAET,OAAO,I; K;IAGX,wC;MAMwB,UACT,M;MAHX,aAAa,aAAa,SAAb,EAAmB,OAAQ,KAA3B,C;MACb,kBAAkB,C;MAC E,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C; ;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,cAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C;M ACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL ,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAW,OAAQ,KAAmB,C;MACb,kBAAkB, C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,W AAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAS,OAAQ,KAAjB,C;MACb,kBAA kB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK ,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,k BAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,U AAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAmB,C;M ACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YA AwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAY,OAAQ,KAAp B,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAA P,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,oBAAa,OAAQ,K AArB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4 BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OA AQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EA AO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OA AO,yBAAY,CAAZ,EAAe,CAAf,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CA AvB,IAA3B,C;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,cAAU,CAAV,C;MAC9B,OAAO,yBAA Y,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;Q AAuB,OAAO,eAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB ,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,eAAS,CAAT,C;MAC9B,OAAO,yBAAY,OA AQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB ,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA 3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAA Q,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB, OAAO,iBAAY,CAAZ,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3 B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,oBAAa,CAAb,C;MAC9B,OAAO,0BAAY,OAAQ, MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OA AO,iBAAU,CAAV,C;MAC9B,OAAO,0BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C ;K;IAGX,4B;MAciB,Q;Mc3nJb,IAAI,EdqnJI,KAAK,CernJT,CAAJ,C;QACI,cdonJc,sD;QcnnJd,MAAM,gCAAYB, OAAQ,WAAjC,C;;MdonJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OA AO,iB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ, WAAW,iBAAa,CAAb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,I AAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McjpJb,IAAI,Ed2oJI,KAAK,Cc3oJT,CA

AJ,C;QACI,cd0oJc,sD;QczoJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md0oJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAGB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McvqJb,IAAI,EdiqJI,KAAK,CcjqJT,CAAJ,C;QACI,cdgqJc,sD;Qc/pJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdgqJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7rJb,IAAI,EdurJI,KAAK,CcvtJT,CAAJ,C;QACI,cdsrJc,sD;QcrrJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdsrJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAae,CAAF,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7sJb,IAAI,Ed6sJI,KAAK,Cc7sJT,CAAJ,C;QACI,cd4sJc,sD;Qc3sJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md4sJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAGB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7uJb,IAAI,EdmuJI,KAAK,CcnuJT,CAAJ,C;QACI,cdkuJc,sD;QcjuJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdkuJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7vJb,IAAI,EdyvJI,KAAK,CczvJT,CAAJ,C;QACI,cdwvJc,sD;QcvvJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdwvJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAkB,CAAIb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7wJb,IAAI,Ed7wJI,KAAK,Cc7wJT,CAAJ,C;QACI,cd8wJc,sD;Qc7wJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md8wJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAmB,CAAnB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7yJb,IAAI,EdqyJI,KAAK,CcryJT,CAAJ,C;QACI,cdoyJc,sD;QcnyJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdoyJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,EAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAGB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,IAAK,WAAI,iBAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,gC;MenzJI,IAAI,Ed2zJI,KAAK,Cc3zJT,CAAJ,C;QACI,cd0zJc,sD;QczzJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md0zJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,iB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAaa,CAAb,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mct0JI,IAAI,Ed80JI,KAAK,Cc90JT,CAAJ,C;QACI,cd60Jc,sD;Qc50Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md60JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAGB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcz1JI,IAAI,Edi2JI,KAAK,Ccj2JT,CAAJ,C;QACI,cdg2Jc,sD;Qc/1Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mdg2JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CA

AJ,C;MACT,OAAO,I;K;IAGX,kC;Mc52JI,IAAI,Edo3JI,KAAK,Ccp3JT,CAAJ,C;QACI,cdm3Jc,sD;Qcl3Jd,MAA  
M,gCAAyB,OAAQ,WAAjC,C;;Mdm3JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,I  
AAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CA  
AP,IAAL,CAAP,C;MACnB,WAAW,iBA Ae,CAAf,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QA  
CI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mc/3JI,IAAI,Edu4JI,KAAK,Ccv4JT,CAAJ,  
C;QACI,cds4Jc,sD;Qcr4Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mds4JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,  
W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,  
OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AgB,CAAhB,C;MACX,iBAAc,OAAO  
,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcl5JI  
,IAAI,Ed05JI,KAAK,Cc15JT,CAAJ,C;QACI,cdy5Jc,sD;Qcx5Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mdy5JV,IA  
AI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MA  
CtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA Ai  
B,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;  
MACT,OAAO,I;K;IAGX,kC;Mcr6JI,IAAI,Ed66JI,KAAK,Cc76JT,CAAJ,C;QACI,cd46Jc,sD;Qc36Jd,MAAM,gCA  
AyB,OAAQ,WAAjC,C;;Md46JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KA  
AK,IAAT,C;QA Ae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAA  
L,CAAP,C;MACnB,WAAW,iBA AkB,CAAiB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IA  
AK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcx7JI,IAAI,Edg8JI,KAAK,Cch8JT,CAAJ,C;Q  
ACI,cd+7Jc,sD;Qc97Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md+7JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;  
MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OA  
AO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AmB,CAAnB,C;MACX,iBAAc,OAAO,C  
AAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mc38JI,I  
AAI,Edm9JI,KAAK,Ccn9JT,CAAJ,C;QACI,cdk9Jc,sD;Qcj9Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mdk9JV,IAA  
I,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACT  
B,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBA AK,OAAO,CAAP,IAAL,EAAP,C;MACnB,WAAW,iBA AgB  
,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBA AK,KAAL,EAJ,C;  
MACT,OAAO,I;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,C  
AA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,  
C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAA  
d,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,C  
AAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iB  
AAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAA  
K,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,u  
C;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,O  
AAO,gBA AK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,g  
D;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL  
,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4  
C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,C  
AAV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,  
8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UA  
AK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAXX,C;kGAcA  
,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,  
UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAX  
X,C;kGAcA,yB;MAAA,8D;MAAA,oC;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/  
B,U;UACI,IAAI,CAAC,UAAU,sBA AK,KAAL,EA AV,CAAL,C;YACI,OAAO,gBA AK,QAAQ,CAAR,IAAL,C;;  
QAGf,OAAO,iB;O;KAXX,C;wFAcA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S

AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S  
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAOiB,Q;QADb,WAA  
W,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YAC  
I,K;UACJ,IAAK,WAAI,iBAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAeA,4B;MAII,eAAe,CAAC,mBAAO,CAAP,IAA  
D,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,wB;MACnB,iBAAc,CAAd,WAAi  
B,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,I  
AAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,  
CAAf,C;QAaKB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAA  
L,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,  
eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,  
0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK  
,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IA  
Aa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QA  
AjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB  
,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,  
C;QAaKB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;Q  
ACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,  
CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,0B;M  
ACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAA  
L,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CA  
Ab,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;  
QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QA  
CrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QA  
AkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,  
UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAA  
C,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAaKB,M;MACIB,mBAAmB,0B;MACnB,  
iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;Q  
ACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,kD;MAWI,oCAaA,2BAaKB,SAaIB,EAA6B,OAA7B,EAAc,  
gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;  
MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;Q  
ACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,kD;MAWI,oCA  
Aa,2BAaKB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CA  
AxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA  
8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,  
IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAaA,2BAaKB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,  
eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAm

B,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAA  
L,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAA  
IB,EAA6B,OAA7B,EAAcC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,  
cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QAC  
I,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,  
mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAAIB,EAA6B,OAA7B,EAAcC,gBAAtC,C;MACb,eAAe,CAAC,YAA  
Y,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;  
MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,Y  
AAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAAIB,EAA6B,OAA7  
B,EAAcC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;  
QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,  
KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;  
MAWI,oCAAA,2BAAkB,SAAIB,EAA6B,OAA7B,EAAcC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,I  
AAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,S  
AAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UA  
AK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAAIB,EAA6B,OAA7B,EAAcC,gBAAt  
C,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3  
B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,U  
AAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2B  
AAkB,SAAIB,EAA6B,OAA7B,EAAcC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;  
MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QA  
A9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAq  
B,G;QACrB,mC;;K;IAIR,6B;MAII,IA+nEO,qBAAQ,CA/nEf,C;QAAe,OAAO,W;MACtB,WAAW,wB;MACN,W  
AAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA6nEO,qBAAQ,CA7nEf,C;QAAe,OAAO,W;MACtB,WAAW,  
0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA2nEO,qBAAQ,CA3nEf,C;QAAe,OAAO,W;M  
ACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAynEO,qBAAQ,CAznEf,C;QAAe  
,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAunEO,qBAAQ,CA  
vnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAqnE  
O,qBAAQ,CAnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B  
;MAII,IAmnEO,qBAAQ,CAnnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO  
,I;K;IAGX,+B;MAII,IAinEO,qBAAQ,CAjnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;M  
ACL,OAAO,I;K;IAGX,+B;MAII,IA+mEO,qBAAQ,CA/mEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WA  
AL,IAAK,C;MACL,OAAO,I;K;IAGX,kC;MAII,IAqiEO,qBAAQ,CARiEf,C;QAAe,OAAO,S;MACtB,aAAa,aAAa,  
SAAb,EAAmB,gBAAnB,C;MACb,gBAAgB,wB;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,C  
AAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAiiEO,qBAAQ,CAjiiEf,C;QAAe,OA  
AO,S;MACtB,aAAa,cAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,  
YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6hEO,qBAAQ,CA7hEf,C;  
QAAe,OAAO,S;MACtB,aAAa,eAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;Q  
ACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAyhEO,qBAA  
Q,CAzhEf,C;QAAe,OAAO,S;MACtB,aAAa,eAAS,gBAAT,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,  
SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqh  
EO,qBAAQ,CARhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CA  
AV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;  
MAII,IAihEO,qBAAQ,CAjhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb  
,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K  
;IAGX,oC;MAII,IA6gEO,qBAAQ,CA7gEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAy,gBAAZ,C;MACb,gBAAgB,  
0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,  
OAAO,M;K;IAGX,oC;MAII,IAygEO,qBAAQ,CAzgEf,C;QAAe,OAAO,S;MACtB,aAAa,oBAAa,gBAAb,C;MAC

b,gBAAgB,0B;MACHB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAIL,IAqgEO,qBAAQ,CArgEf,C;QAAe,OAAO,S;MACTb,aAAa,iBAAU,gBA AV,C;MACb,gBAAgB,0B;MACHB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,U AAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,4B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,qBAAQ,4BAAR,C; K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR, C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR, C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR, C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,sC;MAOI,aAAU,wBAAV,OAA2B,CAA3B,M;QACI,QAAQ, MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL, C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,sC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iB AAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV, UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAA I,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,C AAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,I AAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAA U,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;Q ACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IA IIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WA AW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;M AOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAA K,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAA U,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL, C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAA V,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX ,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,yB;MAAA,oD;MgBn5LA,sC;M AAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB, CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB44Lf,sC;QAMI,IAAI,mBAAO,CAAX,C ;UAAc,oBgB15Ld,eAAW,iBhBk5LsB,QgB15LtB,CAAX,ChBk5Lc,C;;O;KANIB,C;S;GASA,yB;MAAA,oD;MgBz4 LA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,S A+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhBk4Lf,sC;QAMI,IAAI,mBAAO,C AAX,C;UAAc,oBgBx4Ld,eAAW,2BhBw4LgC,QgBx4LhC,CAAX,ChBw4Lc,C;;O;KANIB,C;IASA,mC;MAMI,o BAAS,cAAT,C;K;IAGJ,qC;MAIL,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAIL,IAAI,mBAAO,C AAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAIL,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAIL,IAAI ,mBAAO,CAAX,C;QACI,iB;QACA,oB;;K;IAIR,qC;MAIL,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,q C;MAIL,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAIL,IAAI,mBAAO,CAAX,C;QACI,e;QACA,o B;;K;IAIR,2B;MAMI,OAAqB,OAAAd,sBAAc,C;K;IAGzB,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC, OiB5gMhC,WjB4gMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiBnhMhC,WjBmhM gC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiB1hMhC,WjB0hMgC,C;K;IAG3C,6B;M AI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiBjiMhC,WjBiiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAu B,mB;MAA9B,OAAuC,OiBxiMhC,WjBwiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAu C,OiB/iMhC,WjB+iMgC,C;K;IAG3C,6B;MAI0B,kBAAf,0B;MAAuB,mB;MAA9B,OAAuC,OiBtjMhC,WjBsjMg C,C;K;IAG3C,gC;MAMI,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SepjKiB,Q;MfojKK,mB; MAA7B,OiBhkMO,W;K;IjBmkMX,kC;MAIL,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SeljK iB,Q;MfjkKK,iB;MAA7B,OiBxkMO,W;K;IjB2kMX,kC;MAIL,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MA CD,kBAAd,SehjKiB,Q;MfgjKK,iB;MAA7B,OiBhIMO,W;K;IjBmlMX,kC;MAIL,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAd,Se9iKiB,Q;Mf8iKK,iB;MAA7B,OiBxIMO,W;K;IjB2IMX,kC;MAIL,IA6kDO,qBAA Q,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAAL,SAAK,C;MAAiB,mB;MAA7B,OiBhmMO,W;K;IjBmmMX ,kC;MAIL,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,Se3iKiB,Q;Mf2iKK,iB;MAA7B,OiBxm MO,W;K;IjB2mMX,kC;MAIL,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SeziKiB,Q;MfyiKK,i



B;MAA7B,OiBhnMO,W;K;IjBmnMX,kC;MAIL,IAqlDO,qBAAQ,CARlDf,C;QAAe,OAAO,S;MACD,kBAAT,UAA  
AL,SAAK,C;MAAiB,iB;MAA7B,OiBxnMO,W;K;IjB2nMX,0C;MAMI,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAA  
O,S;MACD,kBAAd,SetnKiB,Q;MfsnKK,sBAAS,cAAT,C;MAA7B,OiBloMO,W;K;IjBqoMX,4C;MAIL,IA2gDO,q  
BAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SeprKiB,Q;MfonKK,6B;MAA7B,OiBloMO,W;K;IjB6oMX,4  
C;MAIL,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SelnKiB,Q;MfknKK,6B;MAA7B,OiBlpMO  
,W;K;IjBqpMX,4C;MAIL,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SehnKiB,Q;MfgnKK,6B;  
MAA7B,OiB1pMO,W;K;IjB6pMX,4C;MAIL,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAT,UAA  
L,SAAK,C;MAAiB,6B;MAA7B,OiBlqMO,W;K;IjBqqMX,4C;MAIL,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S  
;MACD,kBAAd,Se7mKiB,Q;Mf6mKK,6B;MAA7B,OiB1qMO,W;K;IjB6qMX,4C;MAIL,IA2gDO,qBAAQ,CA3gD  
f,C;QAAe,OAAO,S;MACD,kBAAd,Se3mKiB,Q;Mf2mKK,6B;MAA7B,OiBlrMO,W;K;IjBqrMX,4C;MAIL,IAmh  
DO,qBAAQ,CAnhDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAAL,SAAK,C;MAAiB,6B;MAA7B,OiB1rMO,W;K;Ij  
B6rMX,gD;MAMI,IAy8CO,qBAAQ,CAz8Cf,C;QAAe,OAAO,S;MACD,kBAAd,SexrKiB,Q;MfwrKK,iC;MAA7B  
,OiBpsMO,W;K;sFjBusMX,yB;MAAA,wD;MgB5rMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;U  
AAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C  
;W;S;OA4DI,C;MhBqrMf,sC;QAQI,OAAO,sBgB7rMP,eAAW,iBhB6rMiB,QgB7rMjB,CAAX,ChB6rMO,C;O;KA  
RX,C;wFAWA,yB;MAAA,wD;MgBvsMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;Y  
AAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4  
DI,C;MhBgsMf,sC;QAMI,OAAO,sBgBtsMP,eAAW,iBhBssMiB,QgBtsMjB,CAAX,ChBssMO,C;O;KANX,C;wF  
ASA,yB;MAAA,wD;MgBhtMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAs  
B,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBy  
sMf,sC;QAMI,OAAO,sBgB/sMP,eAAW,iBhB+sMiB,QgB/sMjB,CAAX,ChB+sMO,C;O;KANX,C;wFASA,yB;M  
AAA,wD;MgBztMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAA  
tB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBktMf,sC;QA  
MI,OAAO,sBgBxtMP,eAAW,iBhBwtMiB,QgBxtMjB,CAAX,ChBwtMO,C;O;KANX,C;wFASA,yB;MAAA,wD;  
MgBluMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5D  
,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB2tMf,sC;QAMI,OAAO,  
sBgBjuMP,eAAW,iBhBiuMiB,QgBjuMjB,CAAX,ChBiuMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgB3uMA,s  
C;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4  
DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBouMf,sC;QAMI,OAAO,sBgB1uM  
P,eAAW,iBhB0uMiB,QgB1uMjB,CAAX,ChB0uMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgBpvMA,sC;MAA  
A,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA  
5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB6uMf,sC;QAMI,OAAO,sBgBnvMP,eAAW  
,iBhBmvMiB,QgBnvMjB,CAAX,ChBmvMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgB7vMA,sC;MAAA,oC;M  
AAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,C  
AAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBsvMf,sC;QAMI,OAAO,sBgB5vMP,eAAW,iBhB4v  
MiB,QgB5vMjB,CAAX,ChB4vMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgBtwMA,sC;MAAA,oC;MAAA,uB  
AOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EA  
A2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB+vMf,sC;QAMI,OAAO,sBgBrwMP,eAAW,iBhBqwMiB,Qg  
BrwMjB,CAAX,ChBqwMO,C;O;KANX,C;0GASA,yB;MAAA,wD;MgB5vMA,sC;MAAA,oC;MAAA,iCAOe,yB;  
QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+  
EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhBqvMf,sC;QAMI,OAAO,sBgB3vMP,eAAW,2BhB2vM2B,QgB3vM3B,  
CAAX,ChB2vMO,C;O;KANX,C;4GASA,yB;MAAA,wD;MgBrwMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8  
D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/E  
N,CAA3B,C;W;S;OA+EI,C;MhB8vMf,sC;QAMI,OAAO,sBgBlwMP,eAAW,2BhBkwM2B,QgBlwM3B,CAAX,Ch  
BkwMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MgB5wMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,  
4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3  
B,C;W;S;OA+EI,C;MhBqwMf,sC;QAMI,OAAO,sBgBzwMP,eAAW,2BhBywM2B,QgBzwM3B,CAAX,ChBywMO  
,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MgBnxMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAA



;IAGJ,0D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMb,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMb,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMb,OAAnB,C;K;IA2B0B,oD;MAAA,wB;QAAW,2BAAK,KAAL,C;O;K;IAJzC,mC;MAII,OAAO,qBAAa,gBAAb,EAAMb,gCAAnB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,+BAAU,gBAAV,GAAGB,6BAAhB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOkB,kD;MAAA,wB;QAAW,0BAAK,KAAL,C;O;K;IAJxkC,kC;MAII,OAAO,kCAAY,gBAAZ,GAakB,+BAAIB,C;K;IAOiB,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,kCAAW,gBAAx,GAAiB,8BAAjB,C;K;IAOe,4C;MAAA,wB;QAAW,uBAAK,KAAL,C;O;K;IAJrC,+B;MAII,OAAO,gCAAS,gBAAT,GAAe,4BAAf,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOiB,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,gCAAW,gBAAx,GAAiB,8BAAjB,C;K;wFA2CX,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOx+QnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP8zPA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOv/QnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP60PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOtRnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP41PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOrhRnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP22PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOpiRnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP03PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOlRnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QPw5PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAy,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOjlRnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QPu6PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAmQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aApQoC,WAOqHc,CAAY,OAaz,CAAJ,EAA0B,OAA1B,C;;QApQhB,OASqO,W;O;KAIRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAc,mBAAuB,QAAvB,C;QAoQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aArQuC,WaqQnC,CAAY,OAaz,CAAJ,EAA0B,OAA1B,C;;QArQhB,OAuQO,W;O;KANRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAakB,EAAC,EAAD,C;QAC1B,kBAAc,mBAAwB,QAAxB,C;QAqQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aAtQwC,WAsQpC,CAAY,OAaz,CAAJ,EAA0B,OAA1B,C;;QAtQhB,OAwQO,W;O;KAPRX,C;8FAeA,yB;MAAA,0D;M

AAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,m  
BAAsB,QAA1B,C;QAsQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,aAvQsC,WAuQIC,CAAY,OAAZ,CAA  
J,EAA0B,OAA1B,C;;QAvQhB,OAYQO,W;O;KARRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC  
;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAuB,QAAvB,C;QAUQL,Q;  
QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,aAxQuC,WAwQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAxQ  
hB,OA0QO,W;O;KATRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAA  
Y,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAwB,QAAxB,C;QAwQL,Q;QAAhB,iD;UAAgB,cAAhB  
,e;UACI,WAAAY,aAzQwC,WAYQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAZQhB,OA2QO,W;O;KAVRX,C  
;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,E  
AAd,C;QAC1B,kBAAC,mBAAyB,QAAzB,C;QAYQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,aA1QyC,W  
A0QrC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA1QhB,OA4QO,W;O;KAXRX,C;8FAeA,yB;MAAA,0D;MAA  
A,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBA  
A0B,QAA1B,C;QA0QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,aA3Q0C,WA2QtC,CAAY,OAAZ,CAAJ,  
EAA0B,OAA1B,C;;QA3QhB,OA6QO,W;O;KAZRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA6QA,oC  
;MAAA,gC;MA7QA,yC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAu  
B,QAAvB,C;QA2QL,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAAAY,aA5QuC,WA4QnC,CAAY,oBAAZ,CAAJ,  
EAA0B,oBAA1B,C;;QA5QhB,OA8QO,W;O;KA1RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD  
;QAU1,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA6QL,Q;Q  
AAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,aA9QoC,WA8QhC,CAAY,OAAZ,CAAJ,EA9QiD,cA8QvB,CAAe,OA  
Af,CAA1B,C;;QA9QhB,OA9RO,W;O;KA3RX,C;8FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU  
I,eAAiC,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA+QL,Q;QAAhB  
,iD;UAAgB,cAAhB,e;UACI,WAAAY,aAhRoC,WAgRhC,CAAY,OAAZ,CAAJ,EAhRiD,cAgRvB,CAAe,OAAf,CA  
A1B,C;;QAhRhB,OAkRO,W;O;KA7RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAi  
C,cAAIB,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAiRL,Q;QAAhB,iD;U  
AAgB,cAAhB,e;UACI,WAAAY,aAIrOC,WakRhC,CAAY,OAAZ,CAAJ,EAIRiD,cAkRvB,CAAe,OAAf,CAA1B,C  
;;QAIRhB,OAoRO,W;O;KA/RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAiC,cAAI  
B,YAAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAmRL,Q;QAAhB,iD;UAAgB,  
cAAhB,e;UACI,WAAAY,aApRoC,WaORhC,CAAY,OAAZ,CAAJ,EApRiD,cAoRvB,CAAe,OAAf,CAA1B,C;;QAp  
RhB,OAsoR,W;O;KAjSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAiC,cAAIB,YA  
AY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAqRL,Q;QAAhB,iD;UAAgB,cAAh  
B,e;UACI,WAAAY,aAtRoC,WAsRhC,CAAY,OAAZ,CAAJ,EAtrID,cAsRvB,CAAe,OAAf,CAA1B,C;;QAtRhB,O  
AwRO,W;O;KANsX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAiC,cAAIB,YAAAY,gB  
AAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAUrL,Q;QAAhB,iD;UAAgB,cAAhB,e;UA  
CI,WAAAY,aAxRoC,WAwRhC,CAAY,OAAZ,CAAJ,EAxRiD,cAwRvB,CAAe,OAAf,CAA1B,C;;QAxRhB,OA0R  
O,W;O;KARsX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAiC,cAAIB,YAAAY,gBAAZ,  
CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAYrL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,W  
AAAY,aA1RoC,WA0RhC,CAAY,OAAZ,CAAJ,EA1RiD,cA0RvB,CAAe,OAAf,CAA1B,C;;QA1RhB,OA4RO,W;O  
;KAVsX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAU1,eAAiC,cAAIB,YAAAY,gBAAZ,CAAk  
B,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA2RL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAAY,a  
A5RoC,WA4RhC,CAAY,OAAZ,CAAJ,EA5RiD,cA4RvB,CAAe,OAAf,CAA1B,C;;QA5RhB,OA8RO,W;O;KAZs  
X,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA8RA,oC;MAAA,gC;MA9RA,yD;QAU1,eAAiC,cAAIB,YA  
AY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA6RL,Q;QAAhB,iD;UAAgB,cAAh  
B,0B;UACI,WAAAY,aA9RoC,WA8RhC,CAAY,oBAAZ,CAAJ,EA9RiD,cA8RvB,CAAe,oBAAf,CAA1B,C;;QA9R  
hB,OAgsO,W;O;KA3sX,C;gGAcA,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QA  
CI,WAAAY,aAAI,YAAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB  
,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAAY,aAAI,YAAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;  
MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,W  
AAAY,aAAI,YAAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+C;MAUoB,Q;MAAhB,wBA





ACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;  
UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,  
C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MA  
IiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAA  
P,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,  
C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,  
C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aAC  
A,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;  
MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,C  
AAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBA  
AN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,sBAAK,CAAL,EAAP,C;UAAL,K;;UACa,uBAAL,S  
AAK,C;UAHV,K;;MAAP,W;K;IAOJ,kC;MAII,OAAO,iBA Ae,aAAL,SAAK,CAAf,C;K;IAGX,oC;MAKiB,Q;MA  
Db,WAAW,iBAAGB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAI,IAAJ  
,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,C;MACX,wBAAa,SAAb,gB;QAA  
a,WAAA,SAAb,M;QAAMB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA  
Ae,gBAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,  
I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAGB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M  
;QAAMB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,C;M  
ACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;  
MAKiB,Q;MADb,WAAW,iBAAkB,gBAAlB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAA  
K,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAmB,gBAAnB,C;MACX,wBAA  
a,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;M  
ADb,WAAW,iBAAGB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QAAMB,IAAK,WA  
AI,iBAAJ,C;;MACxB,OAAO,I;K;IAGX,0B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,+BAAa,qBAAiB,YAAY,gBAAZ,CAAjB,  
CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAY,gBAAZ,CAApB,C  
AAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY,gBAAZ,CAArB,CA  
Ab,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAmB,YAAY,gBAAZ,CAAnB,CAA  
b,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAY,gBAAZ,CAApB,CAAb,  
C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY,gBAAZ,CAArB,CAAb,C;  
UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAuB,YAAY,gBAAZ,CAAvB,CAAb,C;U  
AHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA  
L,K;aACA,C;UAAK,aAAM,sBAAK,CAAL,EAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAiB,eAAL,gBAAK,EAAa,GAAb,  
CAAjB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;oFAOJ,yB;MAAA,+D;MAwaA,gD;MAxA,uC;QAMW,kBAA  
U,gB;QAsaD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAvA6B,SAuAlB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,E  
AAO,IAAP,C;;QAxahB,OA0aO,W;O;KAhbX,C;sFASA,yB;MAAA,+D;MA0aA,gD;MA1aA,uC;QAMW,kBAAU,  
gB;QAwaD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAZa6B,SAyAlB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EA  
AO,IAAP,C;;QA1ahB,OA4aO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MA4aA,gD;MA5aA,uC;QAMW,kBAAU,g  
B;QA0aD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WA3a6B,SA2alB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAA  
O,IAAP,C;;QA5ahB,OA8aO,W;O;KApbX,C;sFASA,yB;MAAA,+D;MA8aA,gD;MA9aA,uC;QAMW,kBAAU,gB;

QA4aD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA7a6B,SA6a1B,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA9ahB,OAgbO,W;O;KAtbX,C;sFASA,yB;MAAA,+D;MAgbA,gD;MAhbA,uC;QAMW,kBAAU,gB;QA8aD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA/a6B,SA+alB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhhbB,OAkbO,W;O;KAxbX,C;sFASA,yB;MAAA,+D;MAkbA,gD;MALbA,uC;QAMW,kBAAU,gB;QA9bD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAjb6B,SAib1B,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAlbhB,OAobO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MAobA,gD;MAPbA,uC;QAMW,kBAAU,gB;QAkbD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAnb6B,SAmblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QApbhB,OAsoB,W;O;KA5bX,C;sFASA,yB;MAAA,+D;MAsoA,gD;MATbA,uC;QAMW,kBAAU,gB;QAobD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WArb6B,SAqblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtbhB,OAwbO,W;O;KA9bX,C;sFASA,yB;MAAA,+D;MAwbA,oC;MAAA,gD;MAAA,gC;MAxbA,uC;QAMW,kBAAU,gB;QAsbD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAvb6B,SAub1B,CAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxhbB,OA0bO,W;O;KAhcX,C;sFASA,yB;MAAA,+D;MA0bA,gD;MA1bA,uC;QAUW,kBAAU,gB;QAwbD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,Wazb6B,SAyblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1bhB,OA4bO,W;O;KAtcX,C;kGAaA,yB;MAAA,+D;MA5JA,gD;MATJA,uC;QAYW,kBAAiB,gB;QAqJR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAtJoC,SAsJzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAvJhB,OAyJO,W;O;KArKX,C;oGAeA,yB;MAAA,+D;MAyJA,gD;MAzJA,uC;QAYW,kBAAiB,gB;QAwJR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WazJoC,SAYzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1JhB,OA4JO,W;O;KAxKX,C;oGAeA,yB;MAAA,+D;MA4JA,gD;MA5JA,uC;QAYW,kBAAiB,gB;QA2JR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA5JoC,SA4JzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA7JhB,OA+JO,W;O;KA3KX,C;oGAeA,yB;MAAA,+D;MA+JA,gD;MA/JA,uC;QAYW,kBAAiB,gB;QA8JR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA/JoC,SA+JzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhKhB,OAkKO,W;O;KA9KX,C;oGAeA,yB;MAAA,+D;MAkKA,gD;MA1KA,uC;QAYW,kBAAiB,gB;QAiKR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAlKoC,SakKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAnKhB,OAqKO,W;O;KAjLX,C;oGAeA,yB;MAAA,+D;MAqKA,gD;MArKA,uC;QAYW,kBAAiB,gB;QAoKR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WArKoC,SAqKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtKhB,OAwKO,W;O;KAplX,C;oGAeA,yB;MAAA,+D;MAwKA,gD;MAxKA,uC;QAYW,kBAAiB,gB;QAUKR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAxKoC,SAwKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAZhB,OA2KO,W;O;KAvLX,C;oGAeA,yB;MAAA,+D;MA2KA,gD;MA3KA,uC;QAYW,kBAAiB,gB;QA0KR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA3KoC,SA2KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA5KhB,OA8KO,W;O;KA1LX,C;oGAeA,yB;MAAA,+D;MA8KA,oC;MAAA,gD;MAAA,gC;MA9KA,uC;QAYW,kBAAiB,gB;QA6KR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,0B;UACI,WA9KoC,SA8KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA/KhB,OAILO,W;O;KA7LX,C;oGAeA,yB;MAAA,+D;MAiLA,gD;MAjLA,uC;QAYW,kBAAiB,gB;QAglR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WajLoC,SAiLzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAlhB,OAoLO,W;O;KAhMX,C;sGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;uGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;





D;MArPA,yC;QASW,kBAAU,oB;QAqPD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAtPoD,WAsPIC,CAAY,OA  
AZ,C;UOx9UP,U;UADP,YP09Ue,WO19UH,WP09UwB,GO19UxB,C;UACL,IAAI,aAAJ,C;YACH,aPw9UuC,gB;  
YAA5B,Wov9UX,aPu9UgC,GOv9UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPo9UA,iB;UACA,IAAK,WAAI,O  
AAJ,C;;QAxPT,OA0PO,W;O;KAnQX,C;sFAYA,yB;MAAA,wE;MA0PA,+D;MA1PA,yC;QASW,kBAAU,oB;QA  
0PD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA3PqD,WA2P3C,CAAY,OAAZ,C;UOz+UP,U;UADP,YP2+Ue,WO  
3+UH,WP2+UwB,GO3+UxB,C;UACL,IAAI,aAAJ,C;YACH,aPy+UuC,gB;YAA5B,Wox+UX,aPw+UgC,GOx+U  
hC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPq+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA7PT,OA+PO,W;O;KAxQ  
X,C;sFAYA,yB;MAAA,wE;MA+PA,+D;MA/PA,yC;QASW,kBAAU,oB;QA+PD,Q;QAAhB,iD;UAAgB,cAAhB,e  
;UACI,UAhQsD,WAgQ5C,CAAY,OAAZ,C;UO1/UP,U;UADP,YP4/Ue,WO5/UH,WP4/UwB,GO5/UxB,C;UACL,  
IAAI,aAAJ,C;YACH,aP0/UuC,gB;YAA5B,Woz/UX,aPy/UgC,GOz/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP  
s/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAIQT,OAoQO,W;O;KA7QX,C;sFAYA,yB;MAAA,wE;MAoQA,+D;M  
ApQA,yC;QASW,kBAAU,oB;QAoQD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UArQuD,WAqQ7C,CAAY,OAAZ,  
C;UO3gVP,U;UADP,YP6gVe,WO7gVH,WP6gVwB,GO7gVxB,C;UACL,IAAI,aAAJ,C;YACH,aP2gVuC,gB;YA  
A5B,WO1gVX,aP0gVgC,GO1gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPugVA,iB;UACA,IAAK,WAAI,OAA  
J,C;;QAvQT,OAYQO,W;O;KAIRX,C;sFAYA,yB;MAAA,wE;MAyQA,oC;MAAA,+D;MAAA,gC;MAzQA,yC;QA  
SW,kBAAU,oB;QAYQD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,UA1QoD,WA0Q1C,CAAY,oBAAZ,C;UO5hV  
P,U;UADP,YP8hVe,WO9hVH,WP8hVwB,GO9hVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4hVuC,gB;YAA5B,WO3  
hVX,aP2hVgC,GO3hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwhVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5  
QT,OA8QO,W;O;KAvRX,C;sFAYA,yB;MAAA,wE;MA8QA,+D;MA9QA,yD;QAUW,kBAAU,oB;QA8QD,Q;Q  
AAhB,iD;UAAgB,cAAhB,e;UACI,UA/QiD,WA+QvC,CAAY,OAAZ,C;UO9iVP,U;UADP,YPgjVe,WOHjVH,WP  
gjVwB,GOHjVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8iVuC,gB;YAA5B,WO7iVX,aP6iVgC,GO7iVhC,EAAS,MA  
AT,C;YACA,e;;YAEA,c;;UP0iVA,iB;UACA,IAAK,WajRyD,cAiRrD,CAAe,OAaf,CAAJ,C;;QAJRT,OAmRO,W  
;O;KA7RX,C;sFAaA,yB;MAAA,wE;MAMRA,+D;MANRA,yD;QAUW,kBAAU,oB;QAmRD,Q;QAAhB,iD;UAA  
gB,cAAhB,e;UACI,UApRiD,WAoRvC,CAAY,OAAZ,C;UOhkVP,U;UADP,YPkkVe,WOlkVH,WPkkVwB,GOlk  
VxB,C;UACL,IAAI,aAAJ,C;YACH,aPkgVuC,gB;YAA5B,WO/jVX,aP+jVgC,GO/jVhC,EAAS,MAAT,C;YACA,e  
;;YAEA,c;;UP4jVA,iB;UACA,IAAK,WAtRyD,cAsRrD,CAAe,OAaf,CAAJ,C;;QAtRT,OAwRO,W;O;KAISX,C;u  
FAaA,yB;MAAA,wE;MAwRA,+D;MAxRA,yD;QAUW,kBAAU,oB;QAwRD,Q;QAAhB,iD;UAAgB,cAAhB,e;U  
ACI,UAzRiD,WAYRvC,CAAY,OAAZ,C;UOllVP,U;UADP,YPolVe,WOplVH,WPolVwB,GOplVxB,C;UACL,IA  
AI,aAAJ,C;YACH,aPklVuC,gB;YAA5B,WOjlVX,aPilVgC,GOjlVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP8kV  
A,iB;UACA,IAAK,WA3RyD,cA2RrD,CAAe,OAaf,CAAJ,C;;QA3RT,OA6RO,W;O;KAvSX,C;uFAaA,yB;MAA  
A,wE;MA6RA,+D;MA7RA,yD;QAUW,kBAAU,oB;QA6RD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA9RiD,W  
A8RvC,CAAY,OAAZ,C;UOpmVP,U;UADP,YPsmVe,WOtmVH,WPsmVwB,GOtmVxB,C;UACL,IAAI,aAAJ,C;  
YACH,aPomVuC,gB;YAA5B,WOnmVX,aPmmVgC,GOnmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPgmVA,i  
B;UACA,IAAK,WAhSyD,cAgSrD,CAAe,OAaf,CAAJ,C;;QAhST,OakSO,W;O;KA5SX,C;uFAaA,yB;MAAA,wE  
;MAkSA,+D;MAISA,yD;QAUW,kBAAU,oB;QAKSD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAnSiD,WAmSvC,  
CAAY,OAAZ,C;UOtmVP,U;UADP,YPwnVe,WOxnVH,WPwnVwB,GOxnVxB,C;UACL,IAAI,aAAJ,C;YACH,aP  
snVuC,gB;YAA5B,WOrnVX,aPqnVgC,GOrnVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPknVA,iB;UACA,IAAK  
,WArSyD,cAqSrD,CAAe,OAaf,CAAJ,C;;QArST,OAuSO,W;O;KAjTX,C;uFAaA,yB;MAAA,wE;MAuSA,+D;M  
AvSA,yD;QAUW,kBAAU,oB;QAUdS,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAXSiD,WAwSvC,CAAY,OAAZ,  
C;UOxoVP,U;UADP,YP0oVe,WO1oVH,WP0oVwB,GO1oVxB,C;UACL,IAAI,aAAJ,C;YACH,aPwoVuC,gB;YA  
A5B,WovoVX,aPuoVgC,GOvoVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPooVA,iB;UACA,IAAK,WA1SyD,cA  
0SrD,CAAe,OAaf,CAAJ,C;;QA1ST,OA4SO,W;O;KAiTX,C;uFAaA,yB;MAAA,wE;MA4SA,+D;MA5SA,yD;QA  
UW,kBAAU,oB;QA4SD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA7SiD,WA6SvC,CAAY,OAAZ,C;UO1pVP,U;  
UADP,YP4pVe,WO5pVH,WP4pVwB,GO5pVxB,C;UACL,IAAI,aAAJ,C;YACH,aP0pVuC,gB;YAA5B,WOzpVX,  
aPypVgC,GOzpVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPspVA,iB;UACA,IAAK,WA/SyD,cA+SrD,CAAe,OA  
Af,CAAJ,C;;QA/ST,OaiTO,W;O;KA3TX,C;uFAaA,yB;MAAA,wE;MAiTA,+D;MAjTA,yD;QAUW,kBAAU,oB;  
QAItd,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAlTiD,WakTvC,CAAY,OAAZ,C;UO5qVP,U;UADP,YP8qVe,  
WO9qVH,WP8qVwB,GO9qVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4qVuC,gB;YAA5B,WO3qVX,aP2qVgC,GO3q

VhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwqVA,iB;UACA,IAAK,WApTyD,cAoTrD,CAAe,OAAf,CAAJ,C;;QApTT,OAsTO,W;O;KAhUX,C;uFAaA,yB;MAAA,wE;MAsTA,oC;MAAA,+D;MAAA,gC;MatTA,yD;QAUW,kBAAU,oB;QAsTD,Q;QAaHb,iD;UAAgB,cAAhB,0B;UACI,UAvTiD,WAuTvC,CAAY,oBAAZ,C;UO9rVP,U;UADP,YPgsVe,WOhsvH,WPgsVwB,GOhsVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8rVuC,gB;YAA5B,WO7rVX,aP6rVgC,GO7rVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP0rVA,iB;UACA,IAAK,WAZTyD,cAyTrD,CAAe,oBAAf,CAAJ,C;;QAZTT,OA2TO,W;O;KArUX,C;wFAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOp5UP,U;UADP,YPs5Ue,Wot5UH,WP5UwB,GOt5UxB,C;UACL,IAAI,aAAJ,C;YACH,aPo5UuC,gB;YAA5B,WOn5UX,aPm5UgC,GOn5UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPg5UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOr6UP,U;UADP,YPu6Ue,Wov6UH,WPu6UwB,GOv6UxB,C;UACL,IAAI,aAAJ,C;YACH,aPq6UuC,gB;YAA5B,WOp6UX,aPo6UgC,GOp6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPi6UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOt7UP,U;UADP,YPw7Ue,Wox7UH,WPw7UwB,GOx7UxB,C;UACL,IAAI,aAAJ,C;YACH,aPs7UuC,gB;YAA5B,WOr7UX,aPq7UgC,GOr7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPk7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOv8UP,U;UADP,YPy8Ue,Woz8UH,WPY8UwB,GOz8UxB,C;UACL,IAAI,aAAJ,C;YACH,aPu8UuC,gB;YAA5B,Wot8UX,aPs8UgC,Got8UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPm8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOx9UP,U;UADP,YP09Ue,Wo19UH,WP09UwB,GO19UxB,C;UACL,IAAI,aAAJ,C;YACH,aPw9UuC,gB;YAA5B,Wov9UX,aPu9UgC,Gov9UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPo9UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOz+UP,U;UADP,YP2+Ue,Wo3+UH,WP2+UwB,GO3+UxB,C;UACL,IAAI,aAAJ,C;YACH,aPy+UuC,gB;YAA5B,Wox+UX,aPw+UgC,GOx+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPq+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UO1/UP,U;UADP,YP4/Ue,Wo5/UH,WP4/UwB,GO5/UxB,C;UACL,IAAI,aAAJ,C;YACH,aP0/UuC,gB;YAA5B,Woz/UX,aPy/UgC,GOz/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPs/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UO3gVP,U;UADP,YP6gVe,Wo7gVH,WP6gVwB,GO7gVxB,C;UACL,IAAI,aAAJ,C;YACH,aP2gVuC,gB;YAA5B,Wo1gVX,aP0gVgC,GO1gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPugVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAhB,UAAgB,SAaHb,O;UACI,UAAU,YAAY,oBAAZ,C;UO5hVP,U;UADP,YP8hVe,Wo9hVH,WP8hVwB,GO9hVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4hVuC,gB;YAA5B,Wo3hVX,aP2hVgC,GO3hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwhVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UO9iVP,U;UADP,YPgjVe,WohjVH,WPgjVwB,GOhjVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8iVuC,gB;YAA5B,Wo7iVX,aP6iVgC,GO7iVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP0iVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOhkVP,U;UADP,YPkkVe,WolkVH,WPkkVwB,GOlkVxB,C;UACL,IAAI,aAAJ,C;YACH,aPkgVuC,gB;YAA5B,Wo/jVX,aP+jVgC,GO/jVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP4jVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UACI,UAAU,YAAY,OAAZ,C;UOllVP,U;UADP,YPolVe,WoplVH,WPolVwB,GOplVxB,C;UACL,IAAI,aAAJ,C;YACH,aPkIVuC,gB;YAA5B,WojlVX,aPilVgC,GOjlVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP8kVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wB

















IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eArLmB,QAqLJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAaA,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAxLe,QAwLP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA9LP,yB;O;KAHJ,C;kFAMA,yB;MA8LA,8D;MAAA,oC;MA9LA,sC;QAGW,sB;;UakMP,IAx4LO,qBAAQ,CAw4Lf,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAtMmB,QAsMJ,CAAS,oBAAT,C;UACf,aAAU,CAAV,OAaA,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAzMe,QAyMP,CAAS,cAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA/MP,yB;O;KAHJ,C;4FAMA,yB;MAAA,8D;MAAA,sC;QAOI,IAhxLO,qBAAQ,CAgxLf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA/xLO,qBAAQ,CA+xLf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9yLO,qBAAQ,CA8yLf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA7zLO,qBAAQ,CA6zLf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA50LO,qBAAQ,CA40Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA31LO,qBAAQ,CA21Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA2LO,qBAAQ,CA02Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAz3LO,qBAAQ,Cay3Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAx4LO,qBAAQ,CAw4Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAaA,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;UAGnB,OAAO,O;O;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MkBhnbA,iB;MIBgnbA,sC;QAEiB,Q;QAFb,IAr+LO,qBAAQ,CAq+Lf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBznbG,MAAO,KlBynbO,QkBznbP,ElBynbiB,CkznbjB,C;;QIB2nbd,OAAO,Q;O;KANBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBtobA,iB;MIBsobA,sC;QAEiB,Q;QAFb,IAN/LO,qBAAQ,CAm/Lf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR

,WkB/obG,MAAO,KIB+obO,QkB/obP,EIB+obiB,CkB/objB,C;;QIBipbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB5pbA,iB;MIB4pbA,sC;QAEiB,Q;QAFb,IAjgMO,qBAAQ,CAigMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBrqbG,MAAO,KIBqqbO,QkBrqbP,EIBqqbiB,CkBqbjB,C;;QIBuqbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB1rbA,iB;MIBkrbA,sC;QAEiB,Q;QAFb,IA/gMO,qBAAQ,CA+gMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB3rbG,MAAO,KIB2rbO,QkB3rbP,EIB2rbiB,CkB3rbjB,C;;QIB6rbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBxsbA,iB;MIBwsbA,sC;QAEiB,Q;QAFb,IA7hMO,qBAAQ,CA6hMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBjtbG,MAAO,KIBitbO,QkBjtbP,EIBitbiB,CkBjtbjB,C;;QIBmtbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB9tbA,iB;MIB8tbA,sC;QAEiB,Q;QAFb,IA3iMO,qBAAQ,CA2iMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBvubG,MAAO,KIBuubO,QkBvubP,EIBuubiB,CkBvubjB,C;;QIByubd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBpvbA,iB;MIBovbA,sC;QAEiB,Q;QAFb,IAzjMO,qBAAQ,CAyjMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB7vbG,MAAO,KIB6vbO,QkB7vbP,EIB6vbiB,CkB7vbjB,C;;QIB+vbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB1wbA,iB;MIB0wbA,sC;QAEiB,Q;QAFb,IAvkMO,qBAAQ,CAukMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBnxbG,MAAO,KIBmxbO,QkBnxbP,EIBmxbiB,CkBnxbjB,C;;QIBqxbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkBhybA,iB;MIBgybA,sC;QAEiB,Q;QAFb,IArlMO,qBAAQ,CAqlMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBzybG,MAAO,KIByybO,QkBzybP,EIByybiB,CkBzybjB,C;;QIB2ybd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBj0bA,iB;MIBi0bA,sC;QAEiB,Q;QAFb,IA3qMO,qBAAQ,CA2qMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB10bG,MAAO,KIB00bO,QkB10bP,EIB00biB,CkB10bjB,C;;QIB40bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBv1bA,iB;MIBu1bA,sC;QAEiB,Q;QAFb,IAzrMO,qBAAQ,CAyrMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBh2bG,MAAO,KIBg2bO,QkBh2bP,EIBg2biB,CkBh2bjB,C;;QIBk2bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB72bA,iB;MIB62bA,sC;QAEiB,Q;QAFb,IAvsMO,qBAAQ,CAusMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBt3bG,MAAO,KIBs3bO,QkBt3bP,EIBs3biB,CkBt3bjB,C;;QIBw3bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBn4bA,iB;MIBm4bA,sC;QAEiB,Q;QAFb,IArtMO,qBAAQ,CAqtMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB54bG,MAAO,KIB44bO,QkB54bP,EIB44biB,CkB54bjB,C;;QIB84bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBz5bA,iB;MIBy5bA,sC;QAEiB,Q;QAFb,IANuMO,qBAAQ,CAmuMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB16bG,MAAO,KIBk6bO,QkB16bP,EIBk6biB,CkB16bjB,C;;QIB06bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB/6bA,iB;MIB+6bA,sC;QAEiB,Q;QAFb,IAjvMO,qBAAQ,CAivMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBx7bG,MAAO,KIBw7bO,QkBx7bP,EIBw7biB,CkBx7bjB,C;;QIB07bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB8bA,iB;MIBq8bA,sC;QAEiB,Q;QAFb,IA/vMO,qBAAQ,CA+vMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB98bG,MAAO,KIB88bO,QkB98bP,EIB88biB,CkB98bjB,C;;QIBg9bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB39bA,iB;MIB29bA,sC;QAEiB,Q;QAFb,IA7wMO,qBAAQ,CA6wMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBp+bG,MAAO,KIBo+bO,QkBp+bP,EI

Bo+biB,Ckbp+bjB,C;;QIBs+bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkBj/bA,  
iB;MIBi/bA,sC;QAeiB,Q;QAFb,IA3xMO,qBAAQ,CA2xMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,C  
AAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkB1/bG  
,MAAO,KIB0/bO,QkB1/bP,ElB0/biB,CkB1/bjB,C;;QIB4/bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MA  
AA,8D;MAAA,sC;QAaiB,Q;QAFb,IA/2MO,qBAAQ,CA+2Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,  
CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2B  
AAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;M  
AAA,sC;QAaiB,Q;QAFb,IA73MO,qBAAQ,CA63Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,C  
AAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C  
AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;  
QAaiB,Q;QAFb,IA34MO,qBAAQ,CA24Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA  
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q  
;QAFb,IAz5MO,qBAAQ,CAy5Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;  
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA  
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,I  
Av6MO,qBAAQ,CAu6Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a  
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAA  
W,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAR7MO,q  
BAAQ,CAq7Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAA  
V,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA  
GnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAN8MO,qBAAQ,C  
Am8Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U  
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OA  
AO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAj9MO,qBAAQ,CAi9Mf,C;  
UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ  
,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K  
AnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA/9MO,qBAAQ,CA+9Mf,  
C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA  
AQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;  
O;KAnBX,C;4FAsBA,yB;MAAA,8D;MkBlscA,iB;MIBkscA,sC;QAaiB,Q;QAFb,IArjNO,qBAAQ,CAqjNf,C;UAA  
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,  
UAAK,CAAL,CAAT,C;UACR,WkBzscG,MAAO,KIByscO,QkBzscP,ElBysciB,CkBzscjB,C;;QIB2scd,OAAO,Q;  
O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkbttaA,iB;MIBstcA,sC;QAaiB,Q;QAFb,IAjkNO,qBAAQ,CAikNf,C;UAA  
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,  
UAAK,CAAL,CAAT,C;UACR,WkB7tcG,MAAO,KIB6tcO,QkB7tcP,ElB6tciB,CkB7tcjB,C;;QIB+tcD,OAAO,Q;O;  
KAjBX,C;8FAoBA,yB;MAAA,8D;MkB1ucA,iB;MIB0ucA,sC;QAaiB,Q;QAFb,IA7kNO,qBAAQ,CA6kNf,C;UAA  
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,  
UAAK,CAAL,CAAT,C;UACR,WkBjvcG,MAAO,KIBivcO,QkBjvcP,ElBivciB,CkBjvcjB,C;;QIBmvcd,OAAO,Q;  
O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB9vcA,iB;MIB8vcA,sC;QAaiB,Q;QAFb,IAzlnO,qBAAQ,CAylnf,C;UA  
Ae,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS  
,UAAK,CAAL,CAAT,C;UACR,WkB7ycG,MAAO,KIBqwcO,QkB7ycP,ElBqwciB,CkB7ycjB,C;;QIBuwcd,OAA  
O,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBlxcA,iB;MIBkxcA,sC;QAaiB,Q;QAFb,IArmNO,qBAAQ,CAqmNf,  
C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,  
SAAS,UAAK,CAAL,CAAT,C;UACR,WkBzxcG,MAAO,KIByxcO,QkBzxcP,ElByxciB,CkBzxcjB,C;;QIB2xcd,O  
AAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBtycA,iB;MIBsycA,sC;QAaiB,Q;QAFb,IAjnNO,qBAAQ,CAin  
f,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA  
Q,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB7ycG,MAAO,KIB6ycO,QkB7ycP,ElB6yciB,CkB7ycjB,C;;QIB+ycd,

OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBlzcA,iB;MIB0zcA,sC;QAaiB,Q;QAFb,IA7nNO,qBAAQ,CA6nNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBJ0cG,MAAO,KIBi0cO,QkBJ0cP,EiBi0ciB,CkBJ0cjB,C;;QIBm0cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB90cA,iB;MIB80cA,sC;QAaiB,Q;QAFb,IAzoNO,qBAAQ,CAYoNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBr1cG,MAAO,KIBq1cO,QkBr1cP,EiBq1ciB,CkBr1cjB,C;;QlBu1cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MkBl2cA,iB;MIBk2cA,sC;QAaiB,Q;QAFb,IArPN O,qBAAQ,CAqpNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBz2cG,MAAO,KIBy2cO,QkBz2cP,EiBy2ciB,Ck Bz2cjB,C;;QIB22cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBJ4cA,iB;MIBi4cA,sC;QAaiB,Q;QAFb,IA zuNO,qBAAQ,CAYuNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU, CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBx4cG,MAAO,KIBw4cO,QkBx4cP,EiBw4ciB, CkBx4cjB,C;;QIB04cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBr5cA,iB;MIBq5cA,sC;QAaiB,Q;QAF b,IArvNO,qBAAQ,CAqvNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aA AU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB55cG,MAAO,KIB45cO,QkB55cP,EiB45c iB,CkB55cjB,C;;QIB85cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBz6cA,iB;MIBy6cA,sC;QAaiB,Q;Q AFb,IAjwNO,qBAAQ,CAiwNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb, aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBh7cG,MAAO,KIBg7cO,QkBh7cP,EiB g7ciB,CkBh7cjB,C;;QIBk7cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB77cA,iB;MIB67cA,sC;QAaiB, Q;QAFb,IA7wNO,qBAAQ,CA6wNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;Q AAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBp8cG,MAAO,KIBo8cO,QkBp8c P,EiBo8ciB,CkBp8cjB,C;;QIBs8cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBj9cA,iB;MIBi9cA,sC;QA aiB,Q;QAFb,IAzxNO,qBAAQ,CAYxNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B ;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBx9cG,MAAO,KIBw9cO,QkBx 9cP,EiBw9ciB,CkBx9cjB,C;;QIB09cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBr+cA,iB;MIBq+cA,sC; QAaiB,Q;QAFb,IAryNO,qBAAQ,CAqyNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF ,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB5+cG,MAAO,KIB4+cO,Qk B5+cP,EiB4+ciB,CkB5+cjB,C;;QIB8+cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBz/cA,iB;MIBy/cA,s C;QAaiB,Q;QAFb,IAjzNO,qBAAQ,CAizNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhgdG,MAAO,KIBggdO, QkBhgdP,EiBggdiB,CkBhgdjB,C;;QIBkgdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB7gdA,iB;MIB6g dA,sC;QAaiB,Q;QAFb,IA7zNO,qBAAQ,CA6zNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C ;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBphdG,MAAO,KIBo hdO,QkBphdP,EiBohdiB,CkBphdjB,C;;QIBshdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MkBj idA,iB;MIBiidA,sC;QAaiB,Q;QAFb,IAz0NO,qBAAQ,CAY0Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,C AAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxidG ,MAAO,KIBwidO,QkBxidP,EiBwidiB,CkBxidjB,C;;QIB0idd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MA AA,sC;QAWiB,Q;QAFb,IA35NO,qBAAQ,CA25Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT, C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX, KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb, IAv6NO,qBAAQ,CAu6Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW, C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAN7NO,qBAAQ,CAM7 Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O ;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA/7NO,qBAAQ,CA+7Nf,C;UAAe,OAAO,I;QA CtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL, CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;

MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA38NO,qBAAQ,CA28Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAv9NO,qBAAQ,CAu9Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA+NO,qBAAQ,CA++Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA/NO,qBAAQ,CA++Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAj100,qBAAQ,CAi1Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA/100,qBAAQ,CA+1Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QA Fb,IA7m00,qBAAQ,CA6mOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QA Ab,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,C AAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAA A,8D;MAAA,kD;QAaiB,Q;QAFb,IA3n00,qBAAQ,CA2nOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C; 0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAzo00,qBAAQ,CAYoOf,C;UAAe,MAAM,6B;Q ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA L,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QA GnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAvp00,qBAAQ,C AupOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,C AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAai B,Q;QAFb,IArq00,qBAAQ,CAqqOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+ B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAa kB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE; MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IANr00,qBAAQ,CAMrOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA I,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAn BX,C;0FAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAjs00,qBAAQ,CAisOf,C;UA Ae,MAAM,6B;QACrB,eAAe,SAAS,sBAak,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S AAS,sBAak,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YA CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IArx00,qB AAQ,CAqxOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB ;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAak C,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;Q AFb,IAjy00,qBAAQ,CAiyOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a







UAAK,CAAL,C;YACR,QArlE,QAqIP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cA  
CV,WAAW,C;;;UAGnB,qBAAO,O;;;QA3IP,yB;O;KAHJ,C;kFAMA,yB;MA2IA,8D;MA3IA,sC;QAGW,sB;;UA+I  
P,IA98PO,qBAAQ,CA88Pf,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAAL,C;UACd,gBAAqB,wB;UA  
CrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAnJmB,QAmJJ,CAAS,OAAT,C;UACf,aAAU  
,CAAV,OAAa,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAAtJe,QAsJP,CAAS,CAAT,C;YACR,IAAI,2BAA  
W,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA5JP,yB;O;KAHJ,C;kFAMA,yB;M  
A4JA,8D;MA5JA,sC;QAGW,sB;;UAgKP,IA79PO,qBAAQ,CA69Pf,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,U  
AAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eApK  
mB,QAoKJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAvKe,  
QAUkP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBA  
AO,O;;;QA7KP,yB;O;KAHJ,C;kFAMA,yB;MA6KA,8D;MA7KA,sC;QAGW,sB;;UAI LP,IA5+PO,qBAAQ,CA4+P  
f,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;  
YAAoB,qBAAO,O;YAAP,uB;;UACpB,eArLmB,QAqLJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;Y  
ACI,QAAQ,UAAK,CAAL,C;YACR,QAxLe,QAwLP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cAC  
I,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA9LP,yB;O;KAHJ,C;kFAMA,yB;MA8LA,8D;MAAA,oC;M  
A9LA,sC;QAGW,sB;;UAKMP,IA3/PO,qBAAQ,CA2/Pf,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAA  
L,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAtMmB,QAsMJ,  
CAAS,oBAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAzMe,QAyMP,C  
AAS,cAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA  
/MP,yB;O;KAHJ,C;4FAMA,yB;MAAA,8D;MAAA,sC;QAOL,IA4PO,qBAAQ,CAM4Pf,C;UAAe,OAAO,I;QACt  
B,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC  
3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SA  
AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA  
pBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOL,IA15PO,qBAAQ,Cak5Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAA  
K,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SA  
AS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;  
UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAu  
BA,yB;MAAA,8D;MAAA,sC;QAOL,IAj6PO,qBAAQ,CAi6Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;Q  
ACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;Q  
ACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2  
BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,  
8D;MAAA,sC;QAOL,IAh7PO,qBAAQ,CAG7Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAq  
B,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,  
CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CA  
AX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAA  
A,sC;QAOL,IA7PO,qBAAQ,CA+7Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,S  
AAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OA  
Aa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,  
C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOL,  
IA98PO,qBAAQ,CA88Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QA  
CrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,  
M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,U  
AAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOL,IA79PO,q  
BAAQ,CA69Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,  
cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,Q  
AAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YA  
CV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOL,IA5+PO,qBAAQ,CA4  
+Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAj

B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,oC;MAAA,sC;QAOI,IA3/PO,qBAAQ,CA2/Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MkB/gfA,iB;MIB+gfA,sC;QAeiB,Q;QAFb,IAxlQO,qBAAQ,CAwlQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBxhfG,MAAO,KIBwhfO,QkBxhfP,EIBwhfiB,CkBxhfjB,C;;QIB0hfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBrifA,iB;MIBqifA,sC;QAeiB,Q;QAFb,IAtmQO,qBAAQ,CAsmQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB9ifG,MAAO,KIB8ifO,QkB9ifP,EIB8ifiB,CkB9ifjB,C;;QIBgffd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB3jfA,iB;MIB2jfA,sC;QAeiB,Q;QAFb,IApnQO,qBAAQ,CAonQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBpkfG,MAAO,KIBokfO,QkBpkfP,EIBokfiB,CkBpkfjB,C;;QIBskfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBjlfA,iB;MIBilfA,sC;QAeiB,Q;QAFb,IAloQO,qBAAQ,CAkoQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB1lfG,MAAO,KIB0lfO,QkB1lfP,EIB0lfiB,CkB1lfjB,C;;QIB4lfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBvmfA,iB;MIBumfA,sC;QAeiB,Q;QAFb,IAhpQO,qBAAQ,CAgpQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhmfG,MAAO,KIBgnfO,QkBhmfP,EIBgnfiB,CkBhmfjB,C;;QIBknfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB7nfA,iB;MIB6nfA,sC;QAeiB,Q;QAFb,IA9pQO,qBAAQ,CA8pQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBtofG,MAAO,KIBsofO,QkBtofP,EIBsofiB,CkBtofjB,C;;QIBwofd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBnfpA,iB;MIBmpfA,sC;QAeiB,Q;QAFb,IA5qQO,qBAAQ,CA4qQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB5pfG,MAAO,KIB4pfO,QkB5pfP,EIB4pfiB,CkB5pfjB,C;;QIB8pfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBzqfA,iB;MIByqfA,sC;QAeiB,Q;QAFb,IA1rQO,qBAAQ,CA0rQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB1rfG,MAAO,KIBkrfO,QkB1rfP,EIBkrfiB,CkB1rfjB,C;;QIBorfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkB/rfA,iB;MIB+rfA,sC;QAeiB,Q;QAFb,IAxsQO,qBAAQ,CAwsQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxsfG,MAAO,KIBwsfO,QkBxsfP,EIBwsfiB,CkBxsfjB,C;;QIB0sfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBhufA,iB;MIBgufA,sC;QAeiB,Q;QAFb,IA9xQO,qBAAQ,CA8xQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBzufG,MAAO,KIByufO,QkBzufP,EIByufiB,CkBzufjB,C;;QIB2ufd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBtvfA,iB;MIBsvfA,sC;QAeiB,Q;QAFb,IA5yQO,qBAAQ,CA4yQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB/vfG,MAAO,KIB+vfO,QkB/vfP,EIB+vfB,CkB/vfjB,C;;QIBiwfd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB5wfA,iB;MIB4wfA,sC;QAeiB,Q;QAFb,IA1zQO,qBAAQ,CA0zQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBxrfG,MAAO,KIBqxfO,QkBxrfP,EIBqxfiB,CkBxrfjB,C;;QIBuxfd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBlyfA,iB;MIBkyfA,sC;QAeiB,Q;QAFb,IAx0QO,qBAAQ,CAw0Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB3yfG,MAAO,KIB2yfO,QkB3yfP,EIB2yfiB,CkB3yfjB,C;;QIB6yfd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBzxfA,iB;MIBwzfA,sC;QAeiB,Q;QAFb,IA1lQO,qBAA

Q,CAs1Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBj0fG,MAAO,KIBi0fO,QkBj0fP,ElBi0fiB,CkBj0fjB,C;;QIBm0fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB90fA,iB;MIB80fA,sC;QAeiB,Q;QAFb,IAp2QO,qBAAQ,CAo2Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBv1fG,MAAO,KIBu1fO,QkBv1fP,ElBu1fiB,CkBv1fjB,C;;QIBy1fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBp2fA,iB;MIBo2fA,sC;QAeiB,Q;QAFb,IAI3QO,qBAAQ,CAk3Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB72fG,MAAO,KIB62fO,QkB72fP,ElB62fiB,CkB72fjB,C;;QIB+2fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB13fA,iB;MIB03fA,sC;QAeiB,Q;QAFb,IAh4QO,qBAAQ,CAg4Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn4fG,MAAO,KIBm4fO,QkBn4fP,ElBm4fiB,CkBn4fjB,C;;QIBq4fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkBh5fA,iB;MIBg5fA,sC;QAeiB,Q;QAFb,IA94QO,qBAAQ,CA84Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBz5fG,MAAO,KIBy5fO,QkBz5fP,ElBy5fiB,CkBz5fjB,C;;QIB25fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAI+QO,qBAAQ,CAk+Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAh/QO,qBAAQ,CAg/Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9/QO,qBAAQ,CA8/Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5gRO,qBAAQ,CA4gRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA1hRO,qBAAQ,CA0hRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxiRO,qBAAQ,CAwiRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAAtjRO,qBAAQ,CAsjRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAIpRO,qBAAQ,CAokRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAIIRo,qBAAQ,CAklRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MkBjmgBA,iB;MIBimgBA,sC;QAaiB,Q;QAFb,IAxqRO,qBAAQ,CAwqRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBxmgBG,MAAO,KIBwmgBO,QkBxmgBP,ElBwmgBiB,CkBxmgBjB,C;;QIB0mgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBBrngBA,iB;MIBqngBA,sC;QAaiB,Q;QAFb,IAprRO,qBAAQ,CAorRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB5ngBG,MAAO,KIB4ngBO,QkB5ngBP,ElB4ngBiB,CkB5ngBjB,C;;QIB8ngBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBzogBA,iB;MIByo

gBA,sC;QAaiB,Q;QAFb,IAhsRO,qBAAQ,CAGsRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,  
C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhpgBG,MAAO,KI  
BgpgBO,QkBhpgBP,ElBgpgBiB,CkBhpgBjB,C;;QlBkpgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB7  
pgBA,iB;MlB6pgBA,sC;QAaiB,Q;QAFb,IA5sRO,qBAAQ,CA4sRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAA  
K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBp  
qgBG,MAAO,KlBoqgBO,QkBpqgBP,ElBoqgBiB,CkBpqgBjB,C;;QlBsqgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;  
MAAA,8D;MkBjrgBA,iB;MlBirgBA,sC;QAaiB,Q;QAFb,IAxtRO,qBAAQ,CAwtRf,C;UAAe,OAAO,I;QACtB,eA  
Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,  
C;UACR,WkBxrgBG,MAAO,KlBwrgBO,QkBxrgBP,ElBwrgBiB,CkBxrgBjB,C;;QlB0rgBd,OAAO,Q;O;KAjBX,C  
;8FAoBA,yB;MAAA,8D;MkBrsgBA,iB;MlBqsgBA,sC;QAaiB,Q;QAFb,IApuRO,qBAAQ,CAouRf,C;UAAe,OAA  
O,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,  
CAAL,CAAT,C;UACR,WkB5sgBG,MAAO,KlB4sgBO,QkB5sgBP,ElB4sgBiB,CkB5sgBjB,C;;QlB8sgBd,OAAO,  
Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBztgBA,iB;MlBytgBA,sC;QAaiB,Q;QAFb,IAhvRO,qBAAQ,CAGvRf,  
C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,  
SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhugBG,MAAO,KlBgugBO,QkBhugBP,ElBgugBiB,CkBhugBjB,C;;Ql  
BkugBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB7ugBA,iB;MlB6ugBA,sC;QAaiB,Q;QAFb,IA5vRO,  
qBAAQ,CA4vRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,  
iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBpvgBG,MAAO,KlBovgBO,QkBpvgBP,ElBovgBiB,C  
kBpvgBjB,C;;QlBsvgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MkBjwgBA,iB;MlBiwgBA,s  
C;QAaiB,Q;QAFb,IAxwRO,qBAAQ,CAwwRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxwgBG,MAAO,KlB  
wwgBO,QkBxwgBP,ElBwwgBiB,CkBxwgBjB,C;;QlB0wgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;Mk  
BhygBA,iB;MlBgygBA,sC;QAaiB,Q;QAFb,IA51RO,qBAAQ,CA41Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UA  
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,Wk  
BvygBG,MAAO,KlBuygBO,QkBvygBP,ElBuygBiB,CkBvygBjB,C;;QlByygBd,OAAO,Q;O;KAjBX,C;8FAoBA,y  
B;MAAA,8D;MkBpzgBA,iB;MlBozgBA,sC;QAaiB,Q;QAFb,IAx2RO,qBAAQ,CAw2Rf,C;UAAe,OAAO,I;QACt  
B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C  
AAT,C;UACR,WkB3zgBG,MAAO,KlB2zgBO,QkB3zgBP,ElB2zgBiB,CkB3zgBjB,C;;QlB6zgBd,OAAO,Q;O;KA  
jBX,C;+FAoBA,yB;MAAA,8D;MkBx0gBA,iB;MlBw0gBA,sC;QAaiB,Q;QAFb,IAp3RO,qBAAQ,CAo3Rf,C;UA  
Ae,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS  
,UAAK,CAAL,CAAT,C;UACR,WkB/0gBG,MAAO,KlB+0gBO,QkB/0gBP,ElB+0gBiB,CkB/0gBjB,C;;QlBi1gBd,  
OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB51gBA,iB;MlB41gBA,sC;QAaiB,Q;QAFb,IAh4RO,qBAAQ,  
CAG4Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UAC  
I,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn2gBG,MAAO,KlBm2gBO,QkBn2gBP,ElBm2gBiB,CkBn2g  
BjB,C;;QlBq2gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBh3gBA,iB;MlBg3gBA,sC;QAaiB,Q;QAFb  
,IA54RO,qBAAQ,CA44Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA  
U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBv3gBG,MAAO,KlBu3gBO,QkBv3gBP,ElB  
u3gBiB,CkBv3gBjB,C;;QlBy3gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBp4gBA,iB;MlBo4gBA,sC;  
QAaiB,Q;QAFb,IAx5RO,qBAAQ,CAw5Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC  
F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB34gBG,MAAO,KlB24gB  
O,QkB34gBP,ElB24gBiB,CkB34gBjB,C;;QlB64gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBx5gBA,  
iB;MlBw5gBA,sC;QAaiB,Q;QAFb,IAp6RO,qBAAQ,CAo6Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CA  
AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB/5gBG,  
MAAO,KlB+5gBO,QkB/5gBP,ElB+5gBiB,CkB/5gBjB,C;;QlBi6gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,  
8D;MkB56gBA,iB;MlB46gBA,sC;QAaiB,Q;QAFb,IAh7RO,qBAAQ,CAG7Rf,C;UAAe,OAAO,I;QACtB,eAAe,SA  
AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA  
CR,WkBn7gBG,MAAO,KlBm7gBO,QkBn7gBP,ElBm7gBiB,CkBn7gBjB,C;;QlBq7gBd,OAAO,Q;O;KAjBX,C;+  
FAoBA,yB;MAAA,oC;MAAA,8D;MkBh8gBA,iB;MlBg8gBA,sC;QAaiB,Q;QAFb,IA57RO,qBAAQ,CA47Rf,C;U

AAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBV8gBG,MAAO,KIBu8gBO,QkBv8gBP,ElBu8gBiB,CkBv8gBjB,C;;QlBy8gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9gSO,qBAAQ,CA8gSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1hSO,qBAAQ,CA0hSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtiSO,qBAAQ,CAsiSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAIjSO,qBAAQ,CAkjSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAI9jSO,qBAAQ,CA8jSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1kSO,qBAAQ,CA0kSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAImsO,qBAAQ,CAkmSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9mSO,qBAAQ,CA8mSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IApsSO,qBAAQ,CAosSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAItSO,qBAAQ,CAktSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAhuSO,qBAAQ,CAGuSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA9uSO,qBAAQ,CA8uSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA1wSO,qBAAQ,CA0wSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAxxSO,qBAAQ,CAwxSf,C;UAAe,MAAM,6B;QACrB,eAAe

,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;  
UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO  
,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAtySO,qBAAQ,CAsySf,C;UAA  
e,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAA  
S,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,  
WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;  
QAFb,IApzSO,qBAAQ,CAozSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAak,CAAL,EAAT,C;QACF,+B;Q  
AAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAak,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,  
CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;M  
AAA,kD;QAWiB,Q;QAFb,IAx4SO,qBAAQ,CAw4Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAA  
T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,  
QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB  
;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAp5SO,qBAAQ,CAo5Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAA  
K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,  
UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX  
,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAh6SO,qBAAQ,CAg6Sf,C;UAAe,OAAO,I;QACtB,eA  
Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,  
C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA  
AO,Q;O;KajBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA56SO,qBAAQ,CA46Sf,C;UAAe,OA  
AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA  
K,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,  
C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAx7SO,qBAAQ,CAw7  
Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA  
Q,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;  
YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAp8SO,  
qBAAQ,CAo8Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,  
iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAA  
kC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;  
QAFb,IAh9SO,qBAAQ,CAg9Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAA  
b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CA  
AIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB;MAAA,8D;MAA  
A,kD;QAWiB,Q;QAFb,IA59SO,qBAAQ,CA49Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;  
QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QA  
AR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KajBX,C;sGAoBA,yB;M  
AAA,oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAx+SO,qBAAQ,CAw+Sf,C;UAAe,OAAO,I;QACtB,eAAe,SA  
AS,sBAak,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAak,CAAL,EAAT,C;UA  
CR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;  
O;KajBX,C;IAoBA,8B;MASiB,Q;MAFb,IA1jTO,qBAAQ,CA0jTf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CA  
AL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkB1/hBG,MAAO,KIB0/hBE,  
GkB1/hBF,ElB0/hBO,CkB1/hBP,C;;MIB4/hBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA1kTO,qBAAQ,CA0kTf  
,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,C  
AAL,C;QACR,MkBrhiBG,MAAO,KIBqhiBE,GkBrhiBF,ElBqhiBO,CkBrihBP,C;;MIBuhiBd,OAAO,G;K;IAGX,g  
C;MAOiB,Q;MAFb,IAx1TO,qBAAQ,CAw1Tf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,+B;MA  
Ab,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MA  
EvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA9ITO,qBAAQ,CA8ITf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,  
CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAA  
a,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IApmTO,qBAAQ,CAomTf,C;QAAe,OAAO,I;MA  
CtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MA

AM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA1mTO,qBAAQ,CA0mTf,C;  
QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL  
,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAhnTO,q  
BAAQ,CAgnTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QA  
AQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;M  
ASiB,Q;MAFb,IAxnTO,qBAAQ,CAwnTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a  
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,Mk3miBG,MAAO,KIB2miBE,Gk3miBF,EIB2miBO,Ck  
B3miBP,C;;MIB6miBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAhoTO,qBAAQ,CAgoTf,C;QAAe,OAAO,I;MA  
CTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkBhniB  
G,MAAO,KIBgniBE,GkBhniBF,EIBgniBO,CkBhniBP,C;;MIBkniBd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA9  
nTO,qBAAQ,CA8nTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QA  
CI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,wC;MA  
GI,OAAO,yBAAc,UAAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAA  
d,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,0C;MAGI,  
OAAO,2BAAc,UAAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,  
C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,8C;MAOiB,Q;MAFb,IAlwTO,qBAAQ,CAkwTf,C;QAAe  
,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;Q  
ACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;  
K;IAGX,gD;MAOiB,Q;MAFb,IAxwTO,qBAAQ,CAwwTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;M  
ACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAA  
b,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA9wTO,qBA  
AQ,CA8wTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAA  
Q,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,  
C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IApxTO,qBAAQ,CAoxTf,C;QAAe,OAAO,I;MACTb,UAAU,  
UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,  
GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;M  
AFb,IA1xTO,qBAAQ,CA0xTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAA  
V,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;  
UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhyTO,qBAAQ,CAgyTf,C;QAAe,OAAO,  
I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAA  
I,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,  
gD;MAOiB,Q;MAFb,IAtyTO,qBAAQ,CAstyTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MA  
Ab,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GA  
A6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA5yTO,qBAAQ,CA4yTf,C  
;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAA  
L,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OA  
AO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAlzTO,qBAAQ,CAkzTf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,  
C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,  
cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,yB;MAMI,OAj4TO,qBAAQ,C;  
K;IAo4TnB,2B;MAMI,OA14TO,qBAAQ,C;K;IAq4TnB,2B;MAMI,OA4TO,qBAAQ,C;K;IA4TnB,2B;MAMI,OA  
Ap4TO,qBAAQ,C;K;IAu4TnB,2B;MAMI,OA4TO,qBAAQ,C;K;IAw4TnB,2B;MAMI,OA4TO,qBAAQ,C;K;IAy  
4TnB,2B;MAMI,OA4TO,qBAAQ,C;K;IA04TnB,2B;MAMI,OA4TO,qBAAQ,C;K;IA24TnB,2B;MAMI,OA4T  
O,qBAAQ,C;K;gFA44TnB,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IA  
AI,UAAU,OA4V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;gFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,S  
AAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OA4V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;  
K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OA4V,  
CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,c  
AAA,SAAhB,M;QAAsB,IAAI,UAAU,OA4V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAM





B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA11UO,qBAAQ,CAk1Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAh2UO,qBAAQ,CAg2Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KAnBX,C;gGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA7UO,qBAAQ,CA7Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAaqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp8UO,qBAAQ,CAo8Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9UO,qBAAQ,CA9Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAh+UO,qBAAQ,CAg+Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9+UO,qBAAQ,CA8+Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5/UO,qBAAQ,CA4/Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1gVO,qBAAQ,CA0gVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1gVO,qBAAQ,CA0gVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAtiVO,qBAAQ,CAsiVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;4GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5nVO,qBAAQ,CA4nVf,C;UACI,OAAO,I;QACX,kBAaqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1oVO,qBAAQ,CA0oVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxpVO,qBAAQ,CAwpVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtsVO,qBAAQ,CAksVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhtVO,qBAAQ,CAgtVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAC,CAAd







AAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp3XO,qBAAQ,CAo3Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzhnBO,W;QjB0hnBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp4XO,qBAAQ,CAo4Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjnnBO,W;QjBknnBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp5XO,qBAAQ,CAo5Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzknBO,W;QjB0knBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp6XO,qBAAQ,CAo6Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjmnBO,W;QjBkmnBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp7XO,qBAAQ,CAo7Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBznnBO,W;QjB0nnBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp8XO,qBAAQ,CAo8Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjpnBO,W;QjBkpnBP,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0GAWBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IA5hYO,qBAAQ,CA4hYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzqnBO,W;QjB0qnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAWBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IA5iYO,qBAAQ,CA6iYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBlsnBO,W;QjBmsnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IA9jYO,qBAAQ,CA8jYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiB3tnBO,W;QjB4tnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IAkYO,qBAAQ,CA+kYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBpvnBO,W;QjBqvnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IAjnyO,qBAAQ,CAinYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBtynBO,W;QjBuynBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IAloYO,qBAAQ,CAkoYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiB/znBO,W;QjBg0nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IANpYO,qBAAQ,CAnpYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBx1nBO,W;QjBy1nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;U

Cd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,oC;MAA  
A,gD;QAEI,IapqYO,qBAAQ,CAoqYf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,I  
AAb,C;QAA+B,8B;QAA5C,aiBj3nBO,W;QjBk3nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB  
,WAAjB,EAA8B,sBAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;gGay  
BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAcI,IA5vYO,qBAAQ,CA4vYf,C;UAAe,OAAO,W;QACtB,sBAAqB,  
UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAI,aAAJ,C;QAAxC,aiB14nBO,W;QjB2  
4nBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,M  
AAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;kGAwBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAz  
wYO,qBAAQ,CaywYf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,  
gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiB/5nBO,W;QjBg6nBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,g  
BAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIB  
X,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAtxYO,qBAAQ,CAsxYf,C;UAAe,OAAO,W;QACtB  
,sBAAkB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBp7n  
BO,W;QjBq7nBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB  
,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;  
QAWI,IAAnyYO,qBAAQ,CamyYf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACkC,kBAAr  
B,eAAe,gBAAf,C;QAA6B,sBAAI,aAAJ,C;QAA1C,aiBz8nBO,W;QjB08nBP,iBAAc,CAAd,UAAsB,gBAAtB,U;U  
ACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;  
KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAhzYO,qBAAQ,CAGzYf,C;UAAe,OAAO,W;  
QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3  
C,aiB99nBO,W;QjB+9nBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAA  
L,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;M  
AAA,uC;QAWI,IA7zYO,qBAAQ,CA6zYf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACoC  
,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBn/nBO,W;QjBo/nBP,iBAAc,CAAd,UAAsB,gB  
AAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,O  
AAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IA10YO,qBAAQ,CA00Yf,C;UAAe,  
OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,gBAAlB,C;QAAgC,sBAAI,aAAJ,  
C;QAA7C,aiBxgoBO,W;QjBygoBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAA  
K,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAA  
A,gE;MAAA,uC;QAWI,IAv1YO,qBAAQ,CAu1Yf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C  
;QACsC,kBAAZB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,aiB7hoBO,W;QjB7hoBP,iBAAc,CAAd,U  
AAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;  
QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAWI,IAP  
2YO,qBAAQ,CAo2Yf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,g  
BAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,aiBljoBO,W;QjBmjoBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,g  
BAAc,oBAAU,0BAAV,EAAuB,sBAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;  
KAIBX,C;8GAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAcI,IA57YO,qBAAQ,CA47Yf,C;UAAe,OAAO,W;Q  
ACtB,sBAAqB,UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAI,aAAJ,C;QAAxC,aiB  
1koBO,W;QjB2koBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UA  
AK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;gHAWBA,yB;MAAA,qD;M  
AAA,gE;MAAA,uC;QAYI,IA18YO,qBAAQ,CA08Yf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAA  
IB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiBhmoBO,W;QjBimoBP,iBAAc,CA  
Ad,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,M  
AAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAx9  
YO,qBAAQ,CAw9Yf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gB  
AAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBtnoBO,W;QjBunoBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBA  
Ac,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAA  
O,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA+YO,qBAAQ,CAs+Yf,C;UAAe,OA

AO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACkC,kBAArB,eAAe,gBAAf,C;QAA6B,sBAAI,aAAJ,C;QA  
A1C,aiB5ooBO,W;QjB6ooBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EA  
A8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAA  
A,qD;MAAA,gE;MAAA,uC;QAYI,IAP/YO,qBAAQ,CAo/Yf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,  
CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiBlqoBO,W;QjBmqoBP,iBAA  
c,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UAC  
d,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,I  
AlgZO,qBAAQ,CAkgZf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,  
gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBxroBO,W;QjByroBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gB  
AAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OA  
AO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAhhZO,qBAAQ,CAghZf,C;UAAe,O  
AAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,gBAAlB,C;QAAgC,sBAAI,aAAJ,C  
;QAA7C,aiB9soBO,W;QjB+soBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB  
,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;M  
AAA,qD;MAAA,gE;MAAA,uC;QAYI,IA9hZO,qBAAQ,CA8hZf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,C  
AAL,CAAIB,C;QACsC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,aiBpuoBO,W;QjBquoBP,i  
BAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;  
UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;M  
AAA,gC;MAAA,uC;QAYI,IA5iZO,qBAAQ,CA4iZf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,  
C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,aiB1voBO,W;QjB2voBP,iBAAc,CA  
Ad,UAAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,KAAV,EAAiB,0BAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;UACd,  
MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;8EAsBA,yB;MA/zBA,gD;MAAA,gE;MA+zBA,gD;QAc  
W,sB;;UA7zBS,Q;UAHhB,IAP0XO,qBAAQ,CAo0Xf,C;YAAe,qBAAO,OA00BH,OA00BG,C;YAAP,uB;;UACqB  
,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+zBzB,OA/zByB,C;UAA5C,aiBj9mBO,W;UjBk9mBP,kB  
A8zBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cA4zBwB,SA5zBV,CAAU,WAAV,EAAuB,OA0vB,C;YACd,  
MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAyzBP,yB;O;KADJ,C;gFAiBA,yB;MAzzBA,gD;MAAA,gE;MAyz  
BA,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP1XO,qBAAQ,CAo1Xf,C;YAAe,qBAAO,OA0zBH,OA1zBG,C;YAAP,  
uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB,OAzzByB,C;UAA5C,aiBz+mBO,W;Uj  
B0+mBP,kBAwzBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cAszBwB,SAtzBV,CAAU,WAAV,EAAuB,OA0  
vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAmzBP,yB;O;KAFJ,C;gFAkBA,yB;MANzBA,gD;MA  
AA,gE;MAmzBA,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP2XO,qBAAQ,CAo2Xf,C;YAAe,qBAAO,OA0zBH,OA  
zBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB,OAyzByB,C;UAA5C,ai  
BjgnBO,W;UjBkgnBP,kBAkzBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cAgzBwB,SAhzBV,CAAU,WAAV,  
EAAuB,OA0vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA6yBP,yB;O;KAFJ,C;gFAkBA,yB;MA  
7yBA,gD;MAAA,gE;MA6yBA,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP3XO,qBAAQ,CAo3Xf,C;YAAe,qBAAO,  
OA8yBH,OA9yBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6yBzB,OA7yBy  
B,C;UAA5C,aiBzhnBO,W;UjB0hnBP,kBA4yBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cA0yBwB,SA1yBV,  
CAAU,WAAV,EAAuB,OA0vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAuyBP,yB;O;KAFJ,C;g  
FAkBA,yB;MAvyBA,gD;MAAA,gE;MAuyBA,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP4XO,qBAAQ,CAo4Xf,C;Y  
AAe,qBAAO,OAwyBH,OAxyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuy  
BzB,OAvyByB,C;UAA5C,aiBjjnBO,W;UjBkjjBP,kBA5yBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cAoyBw  
B,SApyBV,CAAU,WAAV,EAAuB,OA0vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAiyBP,yB;  
O;KAFJ,C;gFAkBA,yB;MAjyBA,gD;MAAA,gE;MAiyBA,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP5XO,qBAAQ,C  
Ao5Xf,C;YAAe,qBAAO,OAkyBH,OAlyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA  
+B,sBAiyBzB,OAjyByB,C;UAA5C,aiBzknBO,W;UjB0knBP,kBAgyBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YAC  
I,cA8xBwB,SA9xBV,CAAU,WAAV,EAAuB,OA0vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA  
2xBP,yB;O;KAFJ,C;gFAkBA,yB;MA3xB,gD;MAAA,gE;MA2xB,gD;QAeW,sB;;UA7zBS,Q;UAHhB,IAP6XO,  
qBAAQ,CAo6Xf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IA



Ab,C;UAA+B,sBA2xBzB,OA3xByB,C;UAA5C,aiBjmnBO,W;UjBkmnBP,kBA0xBmB,O;UAzxBnB,iD;YAAgB,cAAhB,e;YACI,cAwxBwB,SAxxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAqxBP,yB;O;KafJ,C;gFAkBA,yB;MArxBA,gD;MAAA,gE;MAqxBA,gD;QAeW,sB;;UANxBS,Q;UAHhB,IAp7XO,qBAAQ,CAo7Xf,C;YAAe,qBAAO,OAsxBH,OAtxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAqxzBzB,OArxByB,C;UAA5C,aiBznnBO,W;UjB0nnBP,kBAoxBmB,O;UANxBnB,iD;YAAgB,cAAhB,e;YACI,cAkxBwB,SAIxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA+wBP,yB;O;KafJ,C;gFAkBA,yB;MA/wBA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MA+wBA,gD;QAeW,sB;;UA7wBS,Q;UAHhB,IAp8XO,qBAAQ,CAo8Xf,C;YAAe,qBAAO,OAGxBH,OAhxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+wBzB,OA/wByB,C;UAA5C,aiBjpnBO,W;UjBkpnBP,kBA8wBmB,O;UA7wBnB,iD;YAAgB,cAAhB,0B;YACI,cA4wBwB,SA5wBV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAYwBP,yB;O;KafJ,C;4FAkBA,yB;MAzwBA,gD;MAAA,gE;MAywBA,gD;QAeW,6B;;UA1wBP,IA5hYO,qBAAQ,CA4hYf,C;YAAe,4BAAO,OA0wBI,OA1wBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAywBIB,OAzwBkB,C;UAA5C,aiBzqnBO,W;UjB0qnBP,kBAwwB0B,O;UAvwB1B,wD;YACI,cAswB+B,SAtwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAmwBP,gC;O;KAfJ,C;8FAkBA,yB;MANwBA,gD;MAAA,gE;MAMwBA,gD;QAgBW,6B;;UApwBP,IA7iYO,qBAAQ,CA6iYf,C;YAAe,4BAAO,OAowBI,OPwBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAmwBIB,OAAnwBkB,C;UAA5C,aiBlnBO,W;UjBmsnBP,kBAkwB0B,O;UAjwB1B,wD;YACI,cAgwB+B,SAhwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA6vBP,gC;O;KAhBJ,C;8FAMBA,yB;MA7vBA,gD;MAAA,gE;MA6vBA,gD;QAgBW,6B;;UA9vBP,IA9jYO,qBAAQ,CA8jYf,C;YAAe,4BAAO,OA8vBI,OA9vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6vBIB,OA7vBkB,C;UAA5C,aiB3tnBO,W;UjB4tnBP,kBA4vB0B,O;UA3vB1B,wD;YACI,cA0vB+B,SA1vBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAuvBP,gC;O;KAhBJ,C;8FAMBA,yB;MAvvBA,gD;MAAA,gE;MAuvBA,gD;QAgBW,6B;;UAxvBP,IA/kYO,qBAAQ,CA+kYf,C;YAAe,4BAAO,OAwwBI,OAxxvBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuvBIB,OAvvBkB,C;UAA5C,aiBpvnBO,W;UjBqvnBP,kBAsvB0B,O;UArvB1B,wD;YACI,cAovB+B,SApvBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAivBP,gC;O;KAhBJ,C;8FAMBA,yB;MAjvBA,gD;MAAA,gE;MAivBA,gD;QAgBW,6B;;UAlvBP,IAhmYO,qBAAQ,CAgmYf,C;YAAe,4BAAO,OAkvBI,OA1vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAivBIB,OAjvBkB,C;UAA5C,aiB7wnBO,W;UjB8wnBP,kBAgvB0B,O;UA/uB1B,wD;YACI,cA8uB+B,SA9uBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA2uBP,gC;O;KAhBJ,C;8FAMBA,yB;MA3uBA,gD;MAAA,gE;MA2uBA,gD;QAgBW,6B;;UA5uBP,IAjnYO,qBAAQ,CAinYf,C;YAAe,4BAAO,OA4uBI,OA5uBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2uBIB,OA3uBkB,C;UAA5C,aiBtynBO,W;UjBuynBP,kBA0uB0B,O;UAzuB1B,wD;YACI,cAuwB+B,SAxuBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAquBP,gC;O;KAhBJ,C;8FAMBA,yB;MARuBA,gD;MAAA,gE;MAquBA,gD;QAgBW,6B;;UAtuBP,IAloYO,qBAAQ,CAkoYf,C;YAAe,4BAAO,OAsubI,OAtuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAquBIB,OAruBkB,C;UAA5C,aiB/znBO,W;UjBg0nBP,kBAouB0B,O;UANuB1B,wD;YACI,cAkuB+B,SAluBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA+tBP,gC;O;KAhBJ,C;8FAMBA,yB;MA/tBA,gD;MAAA,gE;MA+tBA,gD;QAgBW,6B;;UAhuBP,IANpYO,qBAAQ,CampYf,C;YAAe,4BAAO,OAguBI,OAhuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+tBIB,OA/tBkB,C;UAA5C,aiBx1nBO,W;UjBy1nBP,kBA8tB0B,O;UA7tB1B,wD;YACI,cA4tB+B,SA5tBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAYtBP,gC;O;KAhBJ,C;8FAMBA,yB;MAztBA,gD;MAAA,gE;MAAA,oC;MAytBA,gD;QAgBW,6B;;UA1tBP,IApqYO,qBAAQ,CAoqYf,C;YAAe,4BAAO,OA0tBI,OA1tBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAytBIB,OAztBkB,C;UAA5C,aiBj3nBO,W;UjBk3nBP,kBAwtB0B,O;UAvtB1B,wD;YACI,cAstB+B,SAttBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAE

X,4BAAO,M,;;QAmTBP,gC;O;KAhBJ,C;gFAmBA,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB ;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MA DhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAE J,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q ACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAg B,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B;M AOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,C AAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA ,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;M ACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kF AGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SA AS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe, C;QACf,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,YAAO,SAAS,oBAAT,CAAP,I,;;QAEJ, OAAO,G;O;KAVX,C;4FAaA,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB, wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;8FAGX,+B; MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OA AT,C,;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cA AA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;M ACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;8FAG X,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SA AS,OAAT,C,;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QA AgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAA kB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G; K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACIB,wBAAgB,SAAhB,gB;UA AgB,cAAhB,UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C,;;QAEJ,OAAO,G;O;KAVX,C;gFAaA,+B;MAUo B,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C, ;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;kFAGX,+B; MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,O AAT,C,;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,c AAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C; MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;kF AGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO, SAAS,OAAT,C,;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB; QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C,;;MAEX,OAAO,G;K;kFAGX,yB;MAAA,oC;MAAA,gC; MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACpB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UA CI,OAAO,SAAS,oBAAT,C,;;QAEJ,OAAO,G;O;KAbX,C;kFAgBA,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBA AgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;kFAGX,+B; MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT ,CAAP,I,;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cA AA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe, C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G; K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAA O,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB, gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q; MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I,;



enB65qBnD,GmB75qB8D,KAAK,KnB65qB5D,SAAS,OAAT,CmB75qBuE,KAAH,CAAhB,C;;QnB+5qBvD,OA  
AO,G;O;KAdX,C;mFAiBA,yB;MmB1mqBA,+B;MnB0mqBA,sC;QAWoB,Q;QADhB,UmBzmqBqC,eAAW,oBn  
BymqB/B,CmBzmqB+B,CAAX,C;QnB0mqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmB96q  
BmD,enB86qBnD,GmB96qB8D,KAAK,KnB86qB5D,SAAS,OAAT,CmB96qBuE,KAAH,CAAhB,C;;QnB7qBvD  
,OAAO,G;O;KAdX,C;kFAiBA,yB;MmB3nqBA,+B;MnB2nqBA,sC;QAWoB,Q;QADhB,UmB1nqBqC,eAAW,oBn  
B0nqB/B,CmB1nqB+B,CAAX,C;QnB2nqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmB/7qB  
mD,enB+7qBnD,GmB/7qB8D,KAAK,KnB+7qB5D,SAAS,OAAT,CmB/7qBuE,KAAH,CAAhB,C;;QnBi8qBvD,O  
AAO,G;O;KAdX,C;mFAiBA,yB;MmB5oqBA,+B;MnB4oqBA,sC;QAWoB,Q;QADhB,UmB3oqBqC,eAAW,oBnB  
2oqB/B,CmB3oqB+B,CAAX,C;QnB4oqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBh9qBm  
D,enBg9qBnD,GmBh9qB8D,KAAK,KnBg9qB5D,SAAS,OAAT,CmBh9qBuE,KAAH,CAAhB,C;;QnBk9qBvD,O  
AAO,G;O;KAdX,C;mFAiBA,yB;MmB7pqBA,+B;MnB6pqBA,sC;QAWoB,Q;QADhB,UmB5pqBqC,eAAW,oBnB  
4pqB/B,CmB5pqB+B,CAAX,C;QnB6pqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBj+qBm  
D,enBi+qBnD,GmBj+qB8D,KAAK,KnBi+qB5D,SAAS,OAAT,CmBj+qBuE,KAAH,CAAhB,C;;QnBm+qBvD,OA  
AO,G;O;KAdX,C;mFAiBA,yB;MmB9qqBA,+B;MnB8qqBA,sC;QAWoB,Q;QADhB,UmB7qqBqC,eAAW,oBnB6  
qqB/B,CmB7qqB+B,CAAX,C;QnB8qqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBl/qBmD,  
enBk/qBnD,GmBl/qB8D,KAAK,KnBk/qB5D,SAAS,OAAT,CmBl/qBuE,KAAH,CAAhB,C;;QnBo/qBvD,OAAO,  
G;O;KAdX,C;kFAiBA,yB;MmB/rqBA,+B;MnB+rqBA,sC;QAWoB,Q;QADhB,UmB9rqBqC,eAAW,oBnB8rqB/B,  
CmB9rqB+B,CAAX,C;QnB+rqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBngrBmD,enBmg  
rBnD,GmBngrB8D,KAAK,KnBmgrB5D,SAAS,OAAT,CmBngrBuE,KAAH,CAAhB,C;;QnBqgrBvD,OAAO,G;O;  
KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MmBhtqBA,+B;MnBgtqBA,sC;QAWoB,Q;QADhB,UmB/sqBqC,e  
AAW,oBnB+sqB/B,CmB/sqB+B,CAAX,C;QnBgtqBrC,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O  
;UACI,MmBphrBmD,enBohrBnD,GmBphrB8D,KAAK,KnBohrB5D,SAAS,oBAAT,CmBphrBuE,KAAH,CAAhB,  
C;;QnBshrBvD,OAAO,G;O;KAdX,C;IAiBA,mC;MAIoB,UAMT,M;MANP,wBAAGB,SAAhB,gB;QAAGB,cAAA,  
SAAhB,M;QACI,IAAI,eAAJ,C;UACI,MAAM,gCAAYB,2BAAwB,SAAXB,MAAZB,C;;MAId,OAAO,0D;K;wFA  
GX,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAh  
B,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAA  
O,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6  
B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAh  
B,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,  
OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;  
QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,  
OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EA  
AY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,  
aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KA  
AM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0  
FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,  
SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,  
MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MA  
AA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,  
SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;Q  
AGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB  
,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAA  
U,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,  
EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QAC  
Z,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,K  
AAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,  
C;0FAoBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,a  
AAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;Y

ACI,KAAM,WAAI,oBAAJ,C;;YAEN,MAAO,WAAI,oBAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAA Y,MAAZ,C;O; KAjBX,C;IAoBA,+B;MAkGI,WkB3orBO,MAAO,KIB2orBG,gBkB3orBH,ElBjrbh,KA2FkB,OkB3orBf,C;MIB4 orBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9FqB,GA8FP,UAAK,C AAL,CA9FO,EAAnB,KA8FqB,CAAM,CAAN,CA9FF,CA8FrB,C;;MA9FT,OAGGO,I;K;IA7FX,iC;MAwGI,WkB 3prBO,MAAO,KIB2prBG,gBkB3prBH,ElB0jrbh,KAiGkB,OkB3prBf,C;MIB4prBd,WAAW,iBAAa,IAAb,C;MA CX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApGqB,GAoGP,UAAK,CAAL,CAPGO,EAAnB,KAoGqB,C AAM,CAAN,CAPGF,CAoGrB,C;;MAPGT,OASGO,I;K;IANGX,iC;MA8GI,WkB3qrBO,MAAO,KIB2qrBG,gBkB3 qrBH,ElBokrbh,KAuGkB,OkB3qrBf,C;MIB4qrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAA I B,M;QACI,IAAK,WA1GqB,GA0GP,UAAK,CAAL,CA1GO,EAAnB,KA0GqB,CAAM,CAAN,CA1GF,CA0GrB, C;;MA1GT,OA4GO,I;K;IAzGX,iC;MAoHI,WkB3rrBO,MAAO,KIB2rrBG,gBkB3rrBH,ElB8krBH,KA6GkB,OkB 3rrBf,C;MIB4rrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAhHqB,GAG HP,UAAK,CAAL,CAhHO,EAAnB,KAgHqB,CAAM,CAAN,CAhHF,CAGHrB,C;;MAhHT,OAKHO,I;K;IA/GX,iC ;MA0HI,WkB3srBO,MAAO,KIB2srBG,gBkB3srBH,ElBwlrBH,KAmHkB,OkB3srBf,C;MIB4srBd,WAAW,iBAAa ,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtHqB,GAsHP,UAAK,CAAL,CAtHO,EAAnB, KAsHqB,CAAM,CAAN,CAtHF,CAsHrB,C;;MatHT,OAwhO,I;K;IARHX,iC;MAGII,WkB3trBO,MAAO,KIB2trB G,gBkB3trBH,ElBkmrBH,KAyHkB,OkB3trBf,C;MIB4trBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAk B,IAAIB,M;QACI,IAAK,WA5HqB,GA4HP,UAAK,CAAL,CA5HO,EAAnB,KA4HqB,CAAM,CAAN,CA5HF,CA 4HrB,C;;MA5HT,OA8HO,I;K;IA3HX,iC;MAsII,WkB3urBO,MAAO,KIB2urBG,gBkB3urBH,ElB4mrBH,KA+Hk B,OkB3urBf,C;MIB4urBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAlIq B,GakIP,UAAK,CAAL,CAIIO,EAAnB,KakIqB,CAAM,CAAN,CAIIF,CakIrB,C;;MAIIT,OAoIO,I;K;IAjIX,iC;M A4II,WkB3vrBO,MAAO,KIB2vrBG,gBkB3vrBH,ElBsnrBH,KaqlkB,OkB3vrBf,C;MIB4vrBd,WAAW,iBAAa,IA Ab,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAxIqB,GAwIP,UAAK,CAAL,CaxIO,EAAnB,KA wIqB,CAAM,CAAN,CaxIF,CAwIrB,C;;MAXIT,OA0IO,I;K;IAvIX,iC;MAkJI,WkB3wrBO,MAAO,KIB2wrBG,gB kB3wrBH,ElBgorBH,KA2IkB,OkB3wrBf,C;MIB4wrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,I AAI B,M;QACI,IAAK,WA9IqB,GA8IP,sBAAK,CAAL,EA9IO,EAAnB,KA8IqB,CAAM,CAAN,CA9IF,CA8IrB,C; ;MA9IT,OAGJO,I;K;8EA7IX,yB;MAAA,gE;MkBzorBA,iB;MIByorBA,8C;QAQI,WkB3orBO,MAAO,KIB2orBG, gBkB3orBH,ElB2orBS,KAAM,OkB3orBf,C;QIB4orBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IA AIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I; O;KAbX,C;8EAgBA,yB;MAAA,gE;MkBzprBA,iB;MIByprBA,8C;QAQI,WkB3prBO,MAAO,KIB2prBG,gBk B3prBH,ElB2prBS,KAAM,OkB3prBf,C;QIB4prBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB, M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I; O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBzqrBA,iB;MIByqrBA,8C;QAQI,WkB3qrBO,MAAO,KIB2qrBG,gBkB3q rBH,ElB2qrBS,KAAM,OkB3qrBf,C;QIB4qrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;U ACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;K AbX,C;8EAgBA,yB;MAAA,gE;MkBzrrBA,iB;MIByrrBA,8C;QAQI,WkB3rrBO,MAAO,KIB2rrBG,gBkB3rrBH,E lB2rrBS,KAAM,OkB3rrBf,C;QIB4rrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IA AK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C; +EAgBA,yB;MAAA,gE;MkBzsrBA,iB;MIBysrBA,8C;QAQI,WkB3srBO,MAAO,KIB2srBG,gBkB3srBH,ElB2srB S,KAAM,OkB3srBf,C;QIB4srBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,W AAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgB A,yB;MAAA,gE;MkBztrBA,iB;MIBytrBA,8C;QAQI,WkB3trBO,MAAO,KIB2trBG,gBkB3trBH,ElB2trBS,KAAM ,OkB3trBf,C;QIB4trBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAA U,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MA AA,gE;MkBzurBA,iB;MIByurBA,8C;QAQI,WkB3urBO,MAAO,KIB2urBG,gBkB3urBH,ElB2urBS,KAAM,OkB3 urBf,C;QIB4urBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,U AAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA, gE;MkBzvrBA,iB;MIByvrBA,8C;QAQI,WkB3vrBO,MAAO,KIB2vrBG,gBkB3vrBH,ElB2vrBS,KAAM,OkB3vrB f,C;QIB4vrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAA

K,CAAL,CAAV,EAAMb,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MkBzwrBA,iB;MlBywrBA,8C;QAQI,WkB3wrBO,MAAO,KlB2wrBG,gBk3wrBH,ElB2wrBS,KAA M,OkB3wrBf,C;QlB4wrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI, UAAU,sBAAK,CAAL,EA AV,EAAMb,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAGBA,kC; MAqGoB,gB;MAHhB,gBAAGB,gB;MACHb,WAAW,iBkbt3rBJ,MAAO,KlBs3rBsB,wBA5FzB,KA4FyB,EAAwB ,EAAxB,Ckbt3rBtB,ElBs3rBmD,SkBt3rBnD,CIBs3rBH,C;MACX,QAAQ,C;MACQ,OA9FL,KA8FK,W;MAAhB, OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhGqB,GAgGP,UAAK ,UAAL,EAAK,kBAAL,SAhGO,EAGGI,OA hGJ,CAGrB,C;;MAhGT,OAKGO,I;K;IA/FX,kC;MA6GoB,gB;MAHh B,gBAAGB,gB;MACHb,WAAW,iBkx4rBJ,MAAO,KlBw4rBsB,wBApGzB,KAoGyB,EAAwB,EAAxB,Ckx4rB tB,ElBw4rBmD,SkBx4rBnD,CIBw4rBH,C;MACX,QAAQ,C;MACQ,OAtGL,KAsGK,W;MAAhB,OAAgB,cAAhB ,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxGqB,GAwGP,UAAK,UAAL,EAAK, kBAAL,SAxGO,EAwGI,OA xGJ,CAGrB,C;;MAxGT,OA0GO,I;K;IAvGX,kC;MAqHoB,gB;MAHhB,gBAAGB,g B;MACHb,WAAW,iBk15rBJ,MAAO,KlB05rBsB,wBA5GzB,KA4GyB,EAAwB,EAAxB,Ck15rBtB,ElB05rBm D,SkB15rBnD,CIB05rBH,C;MACX,QAAQ,C;MACQ,OA9GL,KA8GK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,y B;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhHqB,GAgHP,UAAK,UAAL,EAAK,kBAAL,SAh HO,EAGHI,OA hHJ,CAGrB,C;;MAhHT,OAKHO,I;K;IA/GX,kC;MA6HoB,gB;MAHhB,gBAAGB,gB;MACHb,W AAW,iBk56rBJ,MAAO,KlB46rBsB,wBApHzB,KAoHyB,EAAwB,EAAxB,Ck56rBtB,ElB46rBmD,SkB56rBn D,CIB46rBH,C;MACX,QAAQ,C;MACQ,OAtHL,KAsHK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAA I,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxHqB,GAwHP,UAAK,UAAL,EAAK,kBAAL,SAxHO,EAwHI,OA xHJ,CAGrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAqIoB,gB;MAHhB,gBAAGB,gB;MACHb,WAAW,iBk97r BJ,MAAO,KlB87rBsB,wBA5HzB,KA4HyB,EAAwB,EAAxB,Ck97rBtB,ElB87rBmD,SkB97rBnD,CIB87rBH,C; MACX,QAAQ,C;MACQ,OA9HL,KA8HK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT ,C;UAAoB,K;QACpB,IAAK,WAhIqB,GAgIP,UAAK,UAAL,EAAK,kBAAL,SAhIO,EAGII,OA hIJ,CAGrB,C;;MA hIT,OAKIO,I;K;IA/HX,kC;MA6IoB,gB;MAHhB,gBAAGB,gB;MACHb,WAAW,iBk9rBJ,MAAO,KlB9rBsB,w BApIzB,KAoIyB,EAAwB,EAAxB,Ck9rBtB,ElB9rBmD,SkB9rBnD,CIB9rBH,C;MACX,QAAQ,C;MACQ, OAtIL,KAsIK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK ,WAxIqB,GAwIP,UAAK,UAAL,EAAK,kBAAL,SAxIO,EAwII,OA xIJ,CAGrB,C;;MAxIT,OA0IO,I;K;IAvIX,kC; MAqJoB,gB;MAHhB,gBAAGB,gB;MACHb,WAAW,iBk1+rBJ,MAAO,KlBk+rBsB,wBA5IzB,KA4IyB,EAAwB, EAAxB,Ck1+rBtB,ElBk+rBmD,Sk1+rBnD,CIBk+rBH,C;MACX,QAAQ,C;MACQ,OA9IL,KA8IK,W;MAAhB, OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhJqB,GAgJP,UAAK, UAAL,EAAK,kBAAL,SAhJO,EAGJI,OA hJJ,CAGrB,C;;MAhJT,OAKJO,I;K;IA/IX,kC;MA6JoB,gB;MAHhB,gBA AgB,gB;MACHb,WAAW,iBkbp/rBJ,MAAO,KlBo/rBsB,wBApJzB,KAoJyB,EAAwB,EAAxB,Ckbp/rBtB,ElBo/r BmD,Skbp/rBnD,CIBo/rBH,C;MACX,QAAQ,C;MACQ,OAtJL,KAsJK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,y B;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxJqB,GAwJP,UAAK,UAAL,EAAK,kBAAL,SAxJ O,EAwJI,OA xJJ,CAGrB,C;;MAxJT,OA0JO,I;K;IAvJX,kC;MAqKoB,gB;MAHhB,gBAAGB,gB;MACHb,WAAW, iBkbtgsBJ,MAAO,KlBsgsBsB,wBA5JzB,KA4JyB,EAAwB,EAAxB,CkbtgsBtB,ElBsgsBmD,SkbtgsBnD,CIBsgsB H,C;MACX,QAAQ,C;MACQ,OA9JL,KA8JK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SA AT,C;UAAoB,K;QACpB,IAAK,WAhKqB,GAgKP,sBAAK,UAAL,EAAK,kBAAL,UAhKO,EAGKI,OA hKJ,CAG KrB,C;;MAhKT,OAKKO,I;K;+EA/JX,yB;MAAA,kF;MAAA,gE;Mkbn3rBA,iB;MlBm3rBA,8C;QAWoB,UAEY, M;QAL5B,gBAAGB,gB;QACHb,WAAW,ekbt3rBJ,MAAO,KlBs3rBsB,wBAAN,KAAM,EAAwB,EAAxB,Ckbt3r BtB,ElBs3rBmD,Skbt3rBnD,CIBs3rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;U ACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EAAq B,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAFX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MkBr4rBA,iB;MlBq4rBA,8C ;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHb,WAAW,ekBx4rBJ,MAAO,KlBw4rBsB,wBAAN,KAAM,EAA wB,EAAxB,Ckx4rBtB,ElBw4rBmD,SkBx4rBnD,CIBw4rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cA AhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK, kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAFX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MkBv5 rBA,iB;MlBu5rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHb,WAAW,ekB15rBJ,MAAO,KlB05rBsB,w

BAAN,KAAM,EAAwB,EAAxB,CkB15rBtB,ElB05rBmD,SkB15rBnD,CIB05rBH,C;QACX,QAAQ,C;QACQ,uB; QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,U AAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,k F;MAAA,gE;MkBz6rBA,iB;MlBy6rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,ekB56rBJ, MAAO,KIB46rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkB56rBtB,ElB46rBmD,SkB56rBnD,CIB46rBH,C;QACX, QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IA AK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+ EakBA,yB;MAAA,kF;MAAA,gE;MkB37rBA,iB;MlB27rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACH B,WAAW,ekB97rBJ,MAAO,KIB87rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkB97rBtB,ElB87rBmD,SkB97rBnD ,CIB87rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C; YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET, OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkB78rBA,iB;MlB68rBA,8C;QAWoB,UAEY,M;QAL5 B,gBAAGB,gB;QACHB,WAAW,ekBh9rBJ,MAAO,KIBg9rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBh9rBtB,El Bg9rBmD,SkBh9rBnD,CIBg9rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,I AAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OA ARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkB/9rBA,iB;MlB+9rBA,8C;QA WoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,ekBl+rBJ,MAAO,KIBk+rBsB,wBAAN,KAAM,EAAwB,E AAXB,CkBl+rBtB,ElBk+rBmD,SkBl+rBnD,CIBk+rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C ;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAA L,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkBj/rBA,iB; MlBi/rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,ekBp/rBJ,MAAO,KIBo/rBsB,wBAAN,K AAM,EAAwB,EAAxB,CkBp/rBtB,ElBo/rBmD,SkBp/rBnD,CIBo/rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OA AGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL ,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE; MAAA,oC;MkBngsBA,iB;MlBmgsBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,ekBtgsBJ,M AAO,KIBsgsBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBtgsBtB,ElBsgsBmD,SkBtgsBnD,CIBsgsBH,C;QACX,QA AQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK, WAAI,UAAU,sBAAK,UAAAL,EAAK,kBAAL,UAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkB A,kC;MAwFI,WkBvmsBO,MAAO,KIBumsBG,gBkBvmsBH,ElBshsBH,KAiFkB,OkBvmsBf,C;MlBwmsBd,WAA W,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApFqB,GAoFP,UAAK,CAAL,CAPF O,EAAnB,KAoFqB,CAAM,CAAN,CAPFF,CAoFrB,C;;MApFT,OAsFO,I;K;IANFX,kC;MA8FI,WkBvnsBO,MAA O,KIBunsBG,gBkBvnsBH,ElBgisBH,KAuFkB,OkBvnsBf,C;MlBwnsBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,C AAV,MAAkB,IAAIB,M;QACI,IAAK,WA1FqB,GA0FP,UAAK,CAAL,CA1FO,EAAnB,KA0FqB,CAAM,CAAN, CA1FF,CA0FrB,C;;MA1FT,OA4FO,I;K;IAzFX,kC;MAoGI,WkBvosBO,MAAO,KIBuosBG,gBkBvosBH,ElB0isB H,KA6FkB,OkBvosBf,C;MlBwosBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IA AK,WAhGqB,GAGP,UAAK,CAAL,CAhGO,EAAnB,KAgGqB,CAAM,CAAN,CAhGF,CAGrB,C;;MAhGT,OA kGO,I;K;IA/FX,kC;MA0GI,WkBvpsBO,MAAO,KIBupsBG,gBkBvpsBH,ElBojsBH,KAmGkB,OkBvpsBf,C;MlBw psBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtGqB,GAsGP,UAAK,C AAL,CAtGO,EAAnB,KAsGqB,CAAM,CAAN,CAtGF,CAsGrB,C;;MATGT,OAwoGO,I;K;IARGX,kC;MAGHI,WkB vqsBO,MAAO,KIBuqsBG,gBkBvqsBH,ElB8jsBH,KAyGkB,OkBvqsBf,C;MlBwqsBd,WAAW,iBAAa,IAAb,C;M ACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5GqB,GA4GP,UAAK,CAAL,CA5GO,EAAnB,KA4GqB, CAAM,CAAN,CA5GF,CA4GrB,C;;MA5GT,OA8GO,I;K;IA3GX,kC;MAshI,WkBvrsBO,MAAO,KIBursBG,gBk BvrsBH,ElBwksBH,KA+GkB,OkBvrsBf,C;MlBwrsBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IA AIB,M;QACI,IAAK,WAlHqB,GakHP,UAAK,CAAL,CAIHO,EAAnB,KAKHqB,CAAM,CAAN,CAIHf,CAkHrB, C;;MAIHT,OAoHO,I;K;IAjHX,kC;MA4HI,WkBvssBO,MAAO,KIBussBG,gBkBvssBH,ElBklsBH,KAQHkB,OkB vssBf,C;MlBwssBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAXHqB,GA wHP,UAAK,CAAL,CAXHO,EAAnB,KAWHqB,CAAM,CAAN,CAXHF,CAWHrB,C;;MAXHT,OA0HO,I;K;IAvHX ,kC;MAkII,WkBvtsBO,MAAO,KIButsBG,gBkBvtsBH,ElB4lsBH,KA2HkB,OkBvtsBf,C;MlBwtsBd,WAAW,iBA

Aa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9HqB,GA8HP,sBAAK,CAAL,EA9HO,EA8HE,YA9HrB,KA8HqB,CAAM,CAAN,EA9HF,CA8HrB,C;;MA9HT,OAgIO,I;K;+EA7HX,yB;MAAA,gE;MkBrmsBA,iB;MIBqmsBA,8C;QAQI,WkBvmsBO,MAAO,KIBumsBG,gBkBvmsBH,ElBumsBS,KAAM,OkBvmsBf,C;QlBwmsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvnsBO,MAAO,KIBunsBG,gBkBvnsBH,ElBunsBS,KAAM,OkBvnsBf,C;QlBwnsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvosBO,MAAO,KIBuosBG,gBkBvosBH,ElBuosBS,KAAM,OkBvosBf,C;QlBwosBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvpsBO,MAAO,KIBupsBG,gBkBvpsBH,ElBupsBS,KAAM,OkBvpsBf,C;QlBwpsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvqsBO,MAAO,KIBuqsBG,gBkBvqsBH,ElBuqsBS,KAAM,OkBvqsBf,C;QlBwqsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvrsBO,MAAO,KIBursBG,gBkBvrsBH,ElBursBS,KAAM,OkBvrsBf,C;QlBwrsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MIBqnsBA,8C;QAQI,WkBvssBO,MAAO,KIBussBG,gBkBvssBH,ElBussBS,KAAM,OkBvssBf,C;QlBwssBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MkBrtsBA,iB;MIBqtsBA,8C;QAQI,WkBvtsBO,MAAO,KIButsBG,gBkBvtsBH,ElButsBS,KAAM,OkBvtsBf,C;QlBwtsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,EA9V,EA9AmB,kBAAM,CAAN,EA9AnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,4F;MAQ8D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGvN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAP,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,WAaf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAP,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,WAaf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAAsC,I;MAGIN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAP,CAAP,C;;YAEP,MAAO,gBAAO,



OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuB,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ0D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MAGxN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ2D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MAG1N,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuB,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAhB,UAAgB,SAAhB,O;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,oBAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,0F;MAQyC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACIN,OAAO,kBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuB,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQmC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQiC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAc,I;MAC5M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuB,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQmC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,



gB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCA  
AO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB  
,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA  
O,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,  
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA  
O,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,S  
AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA  
O,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,S  
AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA  
O,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAMoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAA  
hB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA  
O,IAAvB,GAAgC,MAAM,K;K;IAGjD,wB;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,S  
AAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,  
gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAe,C;MA  
Cf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,0B;MAMo  
B,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,cAAO,OAAP,C;;MAEJ,OAAO,G  
;K;IAGX,0B;MAMoB,Q;MADhB,UAAiB,G;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OA  
AO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAgB,cA  
AA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SA  
AhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C;  
MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,  
Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;;MAEJ,OAA  
O,G;K;IAGX,0B;MAKoB,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,cAAO,O  
AAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAiB,G;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAA  
A,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAKoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,S  
AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;Ia5uuBX,oD;MAQuF,wC;K;IARvF,8C  
ASI,Y;MAAuC,8B;K;IAT3C,gF;4FOOA,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL,  
CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,q  
B;MAOI,OAAO,sBAAL,CAAJ,C;K;IAGX,wC;MAII,IAAI,oCAAJ,C;QACI,OAAO,yBAAS,OAAT,C;MACX,OA  
AO,qBAAQ,OAAR,KAAoB,C;K;IAWG,yC;MAAA,qB;QAAE,MAAM,8BAA0B,iDAA8C,aAA9C,MAA1B,C;O;  
K;IAR1C,qC;MAMI,IAAI,8BAAJ,C;QACI,OAAO,sBAAL,KAAJ,C;MACX,OAAO,6BAAgB,KAAhB,EAAuB,uB  
AAvB,C;K;0FAGX,4B;MAOI,OAAO,sBAAL,KAAJ,C;K;IAGX,2D;MAcqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OA  
AsB,KA4Lf,IAAS,CAAT,IA5Le,KA4LD,IAAS,iBA5LvB,SA4LuB,CAA3B,GA5LI,SA4LkC,aA5LnB,KA4LmB,C  
AAtC,GA5L0B,YA4L4B,CA5LnC,KA4LmC,C;;MA3L7D,IAAI,QAAQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;M  
ACX,eAAe,oB;MACf,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,Y  
AAT,EAAS,oBAAT,OA AJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;K;sGAGX,yB;MAAA,8D;MAAA,i  
D;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,sBAAL,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;K  
APjE,C;IAUA,6C;MAcqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OAAy,YAAL,SAAK,EAAU,KAAV,C;MACHB,IAA  
I,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAA  
c,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OA AJ,C;UACI,OAAO,O;;MAEf,OAAO,I;K;sGAGX,yB;  
MAAA,sD;MAAA,mC;QAOI,OAAy,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;gFAUA,gC;MAOW,sB;;QAU  
HS,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAGB,yB;UAAM,IAvHH,SAuHO,CAAU,OA AV,CAAJ,C;YAAw  
B,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MAxHP,yB;K;wFAGJ,gC;MA2VoB,Q;MADhB,WAAe,I;MACC,2B  
;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IARVc,SAqVV,CAAU,OA AV,CAAJ,C;UACI,OAAO,O;;MATVf,  
OAYVO,I;K;wFAtVX,gC;MAOW,qB;;QAwVP,eAAoB,+BAAa,cAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI,  
cAAc,QAAS,W;UACvB,IA3Vc,SA2VV,CAAU,OA AV,CAAJ,C;YAAwB,oBAAO,O;YAAP,sB;;QAE5B,oBAAO  
,I;;MA7VP,wB;K;IAGJ,6B;MAMQ,kBADE,SACF,Q;QAAW,OAAY,SAAL,SAAK,C;;QAE5B,eAAe,oB;QACf,I  
AAI,CAAC,QAAS,UAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,OAAO,QAAS,O;;K;IAK5B,6B;MAKI,I

AAI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,CAAL,C;K;mFAGX,yB;MAAA,iE; MAAA,uC;QAKoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;Y AAwB,OAAO,O;;QACrD,MAAM,gCAAuB,wDAAvB,C;O;KANV,C;oGASA,yB;MAAA,iE;MAAA,uC;QASW, Q;QAAA,+B;;UAYS,U;UAAA,6B;UAAhB,OAAGB,gBAAhB,C;YAAgB,2B;YACZ,aAbwB,SAaX,CAAU,OAAV ,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIBA,kC;QAAA,iB;UAAmC,MAAM,g CAAuB,mEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB, C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,m C;MAKQ,kBADE,SACF,Q;QACI,IAAI,mBAAJ,C;UACI,OAAO,I;;UAEP,OAAO,sBAAK,CAAL,C;;QAGX,eAA e,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,OAAO,QAAS,O;;K;IAK5B,mC;MAII,OAAW,m BAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGpC,gC;MAIoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;Q AAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;0FAGX,yB;MAAA,8D; MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,sBAAI,KAAJ,CAATc,GAAcD,aAAa,KA Ab,C;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAc,sBAAI,KAAJ,CAATc,G AAaD,I;K;IAGjE,uC;MAMiB,Q;MAFb,IAAI,8BAAJ,C;QAAkB,OAAO,SAAK,eAAQ,OAAR,C;MAC9B,YAA Y,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ, C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;IAGX,uC;MAKI,OAAO,wBAAQ,OAAR,C;K;gGAGX,yB;MA AA,wE;MAAA,uC;QAKiB,Q;QADb,YAA Y,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAA nB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAEJ,OAAO,E;O;KAXX,C;gGAcA,gC; MAKiB,Q;MADb,YAA Y,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,UAAU,IAAV,CAAJ,C;UA CI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;8FAGX,yB;MAAA,wE;MAAA,uC;QAMiB,Q;QAFb,gBAAGB,E;QA ChB,YAA Y,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAAnB,C;UACA,IAAI,UAAU,IAA V,CAAJ,C;YACI,YAA Y,K;UACHB,qB;;QAEJ,OAAO,S;O;KAZX,C;8FAeA,gC;MAII,eAAe,SAAK,sBAAa,cAAb ,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,IAAI,UAAU,QAAS,WAAAnB,CAAJ,C;UACI,OAAO,QAAS,Y;;;MAG xB,OAAO,E;K;IAGX,4B;MASQ,kBADE,SACF,Q;QAAW,OAA Y,QAAL,SAAK,C;;QAEEnB,eAAe,oB;QACf,IAA I,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,WAAW,QAAS,O;QACpB,OAAO,QAAS,U AAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,4B;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB, gBAAvB,C;MACV,OAAO,sBAAK,2BAAL,C;K;iFAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAUo B,UAQT,M;QAVP,WAAe,I;QACf,YAA Y,K;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAA U,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA Y,MAAM,gCAAuB,w DAAvB,C;QAEIB,OAAO,2E;O;KAIBX,C;iFAqBA,yB;MAAA,iE;MAAA,uC;QAQI,eAAe,SAAK,sBAAa,cAAb,C ;QACpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO, O;;QAEEnC,MAAM,gCAAuB,kDAAvB,C;O;KAbV,C;IAGBA,2C;MAOiB,Q;MAHb,IAAI,8BAAJ,C;QAAkB,OAA O,SAAK,mBAAY,OAAZ,C;MAC9B,gBAAGB,E;MACHB,YAA Y,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB; QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAA Y,K;QACHB,qB;;MAEJ,OAAO,S; K;IAGX,2C;MAKI,OAAO,4BAAY,OAAZ,C;K;IAGX,kC;MAOQ,kBADE,SACF,Q;QAAW,OAAW,mBAAJ,GA Ae,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;;QAEvC,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI, OAAO,I;QACX,WAAW,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K ;IAKnB,kC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;6FAGpC,gC;MAOo B,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;U ACI,OAAO,O;;;MAGf,OAAO,I;K;6FAGX,gC;MAMI,eAAe,SAAK,sBAAa,cAAb,C;MACpB,OAAO,QAAS,cAA hB,C;QACI,cAAc,QAAS,W;QACvB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MAEnC,OAAO,I;K;qFA GX,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,IAAI,m BAAJ,C;QACI,MAAM,2BAAuB,sBAAvB,C;MACV,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAJB,C;K;iGAGX ,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAA J,C;QACI,OAAO,I;MACX,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAJB,C;K;IAGX,8B;MAKQ,kBADE,SACF, Q;QAAW,OAA Y,UAAL,SAAK,C;;QAEEnB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAu B,sBAAvB,C;QACV,aAAa,QAAS,O;QACTb,IAAI,QAAS,UAAb,C;UACI,MAAM,gCAAyB,uCAAZB,C;QACV, OAAO,M;;K;IAKnB,8B;MAIiB,IAAN,I;MAAA,QAAM,cAAN,C;aACH,C;UAAK,MAAM,2BAAuB,gBAAvB,C;

aACX,C;UAAK,6BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAYB,iCAAzB,C;;MAHIB,W;K;qFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,qDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,wDAAvB,C;QAEIB,OAAO,6E;O;KafX,C;IAkBA,oC;MAKQ,kBADE,SACF,Q;QAAW,OAAW,mBAAQ,C;AAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;;QAEIC,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAo,C;UACI,OAAO,I;QACX,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,OAAO,I;QACX,OAAO,M;;K;IAKnB,oC;MAII,OA AW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;iGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAoBsC,UAGT,MAHS,EAarB,M;MN7pBb,IAAI,EMooBI,KAAK,CNpoBT,CAAJ,C;QACI,cMmoBc,sD;QNloBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMmoBV,IAAI,MAAK,CAAT,C;QAAy,OA AO,mB;MACnB,Q;MACA,IAAI,oCAAJ,C;QACI,iBAAiB,iBAAO,CAAP,I;QACjB,IAAI,cAAc,CAAIB,C;UACI,OAAO,W;QACX,IAAI,eAAc,CAAIB,C;UACI,OAAO,OAAO,kBAAP,C;QACX,OAAO,iBAAa,UAAb,C;QACP,IAAI,8BAAJ,C;UACI,IAAI,sCAAJ,C;YACOB,qB;YAAtB,iBAAc,CAAd,wB;cACI,IAAK,WAAI,sBAAK,KAAL,C;AAJ,C;;YAEI,wCAAa,CAAb,C;YAAb,OAAa,gBAAb,C;cAAa,wB;cACT,IAAK,WAAI,IAAJ,C;;UAEb,OAAO,I;;QAIx,OAAO,gB;;MAEX,YAAY,C;MACC,6B;MAAb,OAAa,gBAAb,C;QAAa,0B;QACT,IAAI,SAAS,CAAb,C;UAAgB,IAAK,WAAI,MAAJ,C;;UAAe,qB;;MAExC,OAAy,qBAAL,IAAK,C;K;IAGhB,kC;MNNqBI,IAAI,EM2qBI,KAAK,CN3qBT,CAAJ,C;QACI,cM0qBc,sD;QNzqBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MM0qBV,OAAO,kBAAGB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;kGAGX,yB;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,IAAI,CAAC,mBAAL,C;UACI,eAAe,+BAAa,cAAb,C;UACf,OAAO,QAAS,cAAhB,C;YACI,IAAI,CAAC,UAAU,QAAS,WAAAnB,CAAL,C;cACI,OAAO,gBAAK,QAAS,YAAT,GAAuB,CAAvB,IAAL,C;;;QAIInB,OAAO,W;O;KAdX,C;0FAiBa,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YA CL,WAAW,I;;;QAE nB,OAAO,I;O;KafX,C;oFAkBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA2FA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA3FU,SA2FN,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QA3FID,OA4FO,W;O;KAIGX,C;kGASA,yB;MAAA,+D;MA6jCA,wE;MA7jCA,uC;QAQW,kBAAGB,gB;QA4jCV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAhjCT,IAZmC,SAY/B,CAgjCkB,oBAAMb,cAAnB,EAAMb,sBAAnB,UAhjCIB,EAgiC+C,IAhjC/C,CAAJ,C;YAA2C,sBAgjCQ,IAhjCR,C;;QAZ/C,OAcO,W;O;KATBX,C;sGAWA,yB;MAkjCA,wE;MALjCA,oD;QAYjCiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAhjCT,IAAI,UAGjCkB,oBAAMb,cAAnB,EAAMb,sBAAnB,UAhjCIB,EAgiC+C,IAhjC/C,CAAJ,C;YAA2C,sBAgjCQ,IAhjCR,C;;QAE/C,OAAO,W;O;KAXX,C;wGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAMb,gB;QASV,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,YAAJ,C;YAAkB,WAAy,WAAI,OAAJ,C;;QATpD,OAuO,W;O;KAhBX,C;4GASA,4C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAAy,WAAI,OAAJ,C;;MACpD,OAAO,W;K;0FAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA4BH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CA5BS,SA4BR,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,OAAJ,C;;QA5B3D,OA6BO,W;O;KANCX,C;IASA,oC;MAMI,OAAO,6BAAGB,gBAAhB,C;K;IAGX,mD;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAy,WAAI,OAAJ,C;;MACvD,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MACID,OAAO,W;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,Od3wBe,W;;Mc4wBtC,OA A6D,SAAtD,SAAK,iBAAQ,OAAQ,MAAhB,EAAuB,OAAQ,aAAR,GAAuB,CAAvB,IAAvB,CAAID,C;K;IAGjE,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAawB,EAaxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OA AO,W;MActB,WAAW,iBAAa,IAAb,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,8B;MAGBiB,Q;MN51Bb,IAAI,EMo1BI,KAAK,CNp1BT,CAAJ,C;QACI,cMm1Bc,sD;QN11Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMm1BV,IAAI,MAAK,CAAT,C;QAAy,OAAO,

W;MACnB,IAAI,oCAAJ,C;QACI,IAAI,KAAK,cAAT,C;UAAe,OAAO,mB;QACtB,IAAI,MAAK,CAAT,C;UAA  
Y,OAAO,OAAO,mBAAP,C;;MAEvB,YAAy,C;MACZ,WAAW,iBAAa,CAAb,C;MACE,2B;MAAb,OAAa,cAAb,  
C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAy,qBAAL,IAAK,  
C;K;IAGhB,kC;MAeqC,IAGhB,I;Mnt3BjB,IAAI,EM42BI,KAAK,CN52BT,CAAJ,C;QACI,cM22Bc,sD;QN12Bd,  
MAAM,gCAAyB,OAAQ,WAAjC,C;;MM22BV,IAAI,MAAK,CAAT,C;QAAy,OAAO,W;MACnB,WAAW,c;MA  
CX,IAAI,KAAK,IAAT,C;QAe,OAAO,mB;MActB,IAAI,MAAK,CAAT,C;QAAy,OAAO,OAAO,kBAAP,C;M  
ACnB,WAAW,iBAAa,CAAb,C;MACX,IAAI,sCAAJ,C;QACI,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;UA  
CI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;QAEI,sCAAa,OAAO,CAAP,IAAb,C;QAAb,OAAa,cAAb,C;UAAa,sB;  
UACT,IAAK,WAAI,IAAJ,C;;MAEb,OAAO,I;K;kGAGX,yB;MAAA,qD;MAAA,gE;MAAA,gD;MAAA,uC;QA  
MI,IAAI,mBAAJ,C;UACI,OAAO,W;QACX,eAAe,+BAAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,IAAI,C  
AAC,UAAU,QAAS,WAAhB,CAAL,C;YACI,QAAS,O;YACT,mBAAmB,iBAAO,QAAS,YAAhB,I;YACnB,IAAI  
,iBAAGB,CAApB,C;cAAuB,OAAO,W;YACI,kBAA3B,eAAa,YAAb,C;YACH,OAAgB,kBAAhB,C;cACI,sBAAa,  
eAAb,C;YAFR,OH11BD,W;;;QGg2BP,OAAO,iB;O;KApBX,C;0FAuBA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;Q  
ADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K  
;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAoBA,+B;MAII,IAAI,wCAAsB,kBAAQ,CAAI,C;Q  
AAqC,OAAO,mB;MAC5C,WAAW,0B;MACN,WAAI,IAAK,C;MACL,OAAO,I;K;IAGX,uC;MAOI,aAAU,2BA  
AV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,sBAAK,CAAL,EAAU,SAAK,  
aAAI,CAAJ,EAAO,sBAAK,CAAL,CAAP,CAAF,C;;K;oFAIR,yB;MAAA,oD;MJn4BA,sC;MAAA,oC;MAAA,uB  
AOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EA  
A2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MI43Bf,sC;QAMI,IAAI,iBAAO,CAAX,C;UAAc,oBJl4Bd,eAA  
W,iBIk4BsB,QJl4BtB,CAAX,CIk4Bc,C;;O;KANIB,C;wGASA,yB;MAAA,oD;MJz3BA,sC;MAAA,oC;MAAA,iC  
AOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA  
2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MIk3Bf,sC;QAMI,IAAI,iBAAO,CAAX,C;UAAc,oBJx3Bd,eAAW,  
2BIw3BgC,QJx3BhC,CAAX,CIw3Bc,C;;O;KANIB,C;IASA,sC;MAMI,sBAAS,cAAT,C;K;IAGJ,6B;MASgB,Q;M  
AHZ,IAAI,oCAAJ,C;QACI,IAAI,kBAAQ,CAAZ,C;UAAe,OAAy,SAAL,SAAK,C;QAEwB,kBAA3C,sBC5+BsD,  
sBD4+BtD,uB;QAAmD,mB;QAA3D,OAAoE,OH17BjE,WGk7BiE,C;;MAEjD,kBAAhB,0B;MAAwB,oB;MAA/B,  
OHp7BO,W;K;wFGu7BX,yB;MAAA,wD;MJ56BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAA  
A,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;  
S;OA4DI,C;MIq6Bf,sC;QAQI,OAAO,sBJ76BP,eAAW,iBI66BiB,QJ76BjB,CAAX,CI66BO,C;O;KARX,C;4GAW  
A,yB;MAAA,wD;MJp6BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB  
;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MI65Bf,sC;Q  
AMI,OAAO,sBJn6BP,eAAW,2BI66B2B,QJn6B3B,CAAX,CI66BO,C;O;KANX,C;IASA,uC;MAMI,OAAO,wBA  
AW,cAAX,C;K;IAGX,6C;MASE,Q;MAHX,IAAI,oCAAJ,C;QACG,IAAI,kBAAQ,CAAZ,C;UAAe,OAAy,SAAL,  
SAAK,C;QAEe,kBAAlC,sBCvhCuD,sBDuhCvD,uB;QAA0C,iC;QAAID,OAAyE,OH79BtE,WG69BqE,C;;MAEr  
D,kBAAhB,0B;MAAwB,mC;MAA/B,OH/9BO,W;K;IGk+BX,qC;MAMoB,UACL,M;MAHX,aAAa,oBAAa,cAAb  
,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAA  
kB,O;;MActB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,cAAU,cAAV,C;MACb,YAAy,C;MACI,2  
B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MActB,OAAO,M;K;I  
AGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB  
,C;QAAGB,oC;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MActB,OAAO,M;K;IAGX,oC;MAMoB,UACL,  
M;MAHX,aAAa,iBAAy,cAAZ,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OA  
AO,cAAP,EAAO,sBAAP,YAAkB,O;;MActB,OAAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,iBAAW,  
cAAX,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP  
,YAAkB,O;;MActB,OAAO,M;K;IAGX,iC;MAMoB,UACL,M;MAHX,aAAa,eAAS,cAAT,C;MACb,YAAy,C;M  
ACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MActB,OAAO,  
M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,c  
AAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MActB,OAAO,M;K;IAGX,mC;MAMoB,U  
ACL,M;MAHX,aAAa,eAAW,cAAX,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAC

Z,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MAcTb,OAAO,M;K;0FAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;M  
AAA,uE;MAAA,uC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,C;QACjD,k  
BAAY,mBAAoB,QAAPB,C;QAYEH,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA1E8C,SA0  
E/B,CAAU,OAAV,C;UbpkBnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;Qa0fA,OA4EO,W;O;KAXFX,C;  
+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,E  
AAXB,CAAZ,CAAyC,EAAC,EAAD,C;QACjD,kBAAC,mBAAoB,QAAPB,C;QA2BL,Q;QAAA,2B;QAAhB,OAAG  
B,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA5BoC,WA4BhC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA5BhB,O  
A8BO,W;O;KA1CX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAwD,cAAz  
C,YAAY,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,C;QACjD,kBAAC,mBAAoB,QAAPB,C;QA6BL,Q;QA  
AA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA9BoC,WA8BhC,CAAY,OAAZ,CAAJ,EA9BiD,  
cA8BvB,CAAe,OAaf,CAA1B,C;;QA9BhB,OAAGCO,W;O;KA3CX,C;mGAcA,+C;MAUoB,Q;MAAA,2B;MAAhB  
,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,  
W;K;mGAGX,+D;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,YAAY,O  
AAZ,CAAJ,EAA0B,eAAe,OAaf,CAA1B,C;;MAEhB,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,O  
AAGB,cAAhB,C;QAAGB,yB;QACZ,WAAe,UAAU,OAAV,C;QbpkBnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OA  
ARb,C;;MaskBA,OAAO,W;K;kGAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAYI,aAAa,  
mBAA6D,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,CAA7D,C;QAcG,Q;QAAA,2B;QAAh  
B,OAAGB,cAAhB,C;UAAgB,yB;UAbO,MAcP,aAAI,OA AJ,EAde,aAcF,CAAc,OAAD,CAAb,C;;QAdhB,OAAuB,  
M;O;KAb3B,C;sGAgBA,iD;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,  
OAAJ,EAAa,cAAc,OAAD,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;  
QAAa,sB;QACT,WAAy,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAII,OAAO,0BAAa,eAAW,YAAY,m  
CAAwB,EAAXB,CAAZ,CAAX,CAAb,C;K;IAGX,6B;MAKqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,QAAM,c  
AAN,C;eACH,C;YAAK,kB;YAAL,K;eACA,C;YAAK,cAAW,8BAAJ,GAakB,sBAAI,CAAJ,CAAIB,GAA8B,oB  
AAW,OAAhD,C;YAAL,K;;YACa,uBAAL,SAAK,C;YAHV,K;;QAAP,W;;MAMJ,OAA4B,qBAAhB,gBAAL,SA  
AK,CAAGB,C;K;IAGhC,oC;MAII,IAAI,oCAAJ,C;QACI,OAAy,gBAAL,SAAK,C;MACHB,OAAO,0BAAa,gBA  
Ab,C;K;IAGX,oC;MAII,OAAO,iBAAU,SAAV,C;K;IAGX,4B;MAOqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,Q  
AAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAakB,sBAAK,CAAL,CAAIB,GA  
A+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,iCAAa,qBAAiB,YAAY,cAAZ,CAAjB,CAAb,C;YAhL,K;;QAAP,W;;  
MAMJ,OAAwC,oBAAjC,0BAAa,sBAAb,CAAI,C;K;sFAG5C,yB;MAAA,+D;MAwFA,gD;MAxFA,uC;QAMW,  
kBAAU,gB;QASFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA vF6B,SAuFIB,CAAU,OAA  
V,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAxFhB,OA0FO,W;O;KAhGX,C;uFASA,yB;MAAA,+D;MA0FA,g  
D;MA1FA,uC;QAUW,kBAAU,gB;QAwFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAzF6  
B,SAyFIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA1FhB,OA4FO,W;O;KA tGX,C;oGAaA,yB;  
MAAA,+D;MA8BA,wE;MAAA,gD;MA9BA,uC;QAYW,kBAAiB,gB;QA6BR,gB;QADhB,YAAY,C;QACI,2B;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA9BoC,SA8BzB,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UA  
AV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA/BhB,OAiCO,W;O;KA7CX,C;oGAeA,yB;MA  
AA,+D;MAiCA,wE;MAAA,gD;MAjCA,uC;QAYW,kBAAiB,gB;QAGCR,gB;QADhB,YAAY,C;QACI,2B;QAAh  
B,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAjCoC,SAiCzB,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,E  
AAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA1ChB,OAoCO,W;O;KAhDX,C;wGAeA,yB;MAAA,  
wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,  
yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy  
,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B  
,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB  
,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX  
,C;0FAkBA,yB;MAAA,gD;MAAA,oD;QAIOB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAA  
W,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;2FAWA,yB;MAAA,g  
D;MAAA,oD;QAQoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UA  
CC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;uFAeA,yB;MAAA,wE;MAyBA,+D;MAzBA,y

C;QASW,kBAAU,oB;QAYBD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA1BiD,WA0BvC,C  
AAy,OAAZ,C;UbnvCP,U;UADP,YaynCe,WbznCH,WaynCwB,GbznCxB,C;UACL,IAAI,aAAJ,C;YACH,aaunCu  
C,gB;YAA5B,WbznCX,aasnCgC,GbznChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UamnCA,iB;UACA,IAAK,WAAI  
,OAAJ,C;;QA5BT,OA8BO,W;O;KAvCX,C;uFAYA,yB;MAAA,wE;MA8BA,+D;MA9BA,yD;QAUW,kBAAU,oB  
;QA8BD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA/BiD,WA+BvC,CAAY,OAAZ,C;UbzoC  
P,U;UADP,Ya2oCe,Wb3oCH,Wa2oCwB,Gb3oCxB,C;UACL,IAAI,aAAJ,C;YACH,aaYoCuC,gB;YAA5B,WbzoC  
X,aaWoCgC,GbzoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UaqoCA,iB;UACA,IAAK,WAjCyD,cAiCrD,CAAe,O  
AAf,CAAJ,C;;QAJCT,OAmCO,W;O;KA7CX,C;0FAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAA,2B;QAAhB,  
OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UbnvCP,U;UADP,YaynCe,WbznCH,WaynCwB,G  
bznCxB,C;UACL,IAAI,aAAJ,C;YACH,aaunCuC,gB;YAA5B,WbznCX,aasnCgC,GbznChC,EAAS,MAAT,C;YAC  
A,e;;YAEA,c;;UamnCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KADx,C;2FAiBA,yB;MAAA,+D;  
MAAA,sE;QAUoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UbzoC  
P,U;UADP,Ya2oCe,Wb3oCH,Wa2oCwB,Gb3oCxB,C;UACL,IAAI,aAAJ,C;YACH,aaYoCuC,gB;YAA5B,WbzoC  
X,aaWoCgC,GbzoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UaqoCA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,  
C;;QAET,OAAO,W;O;KAFx,C;4FAkBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,  
oDASQ,Y;QAA6C,OAAA,oBAAgB,W;O;MATrE,iDAUQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,  
yC;QAQL,2D;O;KARJ,C;8EAca,yB;MAAA,kF;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,mCAAwB,EAAXB  
,CAAb,C;QAuEA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAxEwC,SAwEpC,CAAU,IAA  
V,CAAJ,C;;QAxEhB,OAYEO,W;O;KAhFX,C;4FAUA,yB;MAAA,kF;MAAA,gE;MA+BA,wE;MA/BA,uC;QAO  
W,kBAaA,eAAa,mCAAwB,EAAXB,CAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAA  
a,sB;UACT,WAAy,WAjC+C,SAiC3C,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,IAAvC,CAAJ  
,C;;QAJChB,OAKCO,W;O;KAZCX,C;0GAUA,yB;MAAA,+D;MAoSA,wE;MApSA,uC;QAOW,kBAAoB,gB;QAO  
Sd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1RsB,U;UAAA,cAVQ,SAUR,CAORT,oB  
AAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W;YAA6C,6B;;;QAVhF,OAWO,W;O;KAlBX,C;8G  
AUA,yB;MA0RA,wE;MA1RA,oD;QAIiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1  
RsB,U;UAAA,wBA0RT,oBAAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W;YAA6C,6B;;;QACHF,  
OAAO,W;O;KARX,C;+FAWA,yB;MAAA,wE;MAAA,oD;QAQiB,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAA  
b,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAAI,UAAU,oBAAMb,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,IA  
AvC,CAAJ,C;;QACHB,OAAO,W;O;KAVX,C;4FAaA,yB;MAAA,+D;MAAA,uC;QAOW,kBAAa,gB;QAwPJ,Q;Q  
AAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAhPK,U;UAAA,cARe,SAQf,CAgPQ,OAHPR,W;YAA5C,6B;;;  
QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAqPoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAg  
B,yB;UAhPK,U;UAAA,wBAgPQ,OAHPR,W;YAA5C,6B;;;QAC3D,OAAO,W;O;KANX,C;kFASA,6C;MAKiB,Q;  
MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;  
IAQiB,4C;MAAA,mB;QAAE,gC;O;K;IAL9B,gC;MAKI,OAAO,qBAaiB,6BAajB,C;K;IAGX,+B;MASI,OAA2B,  
SAAf,eAAL,SAAK,CAAe,C;K;4FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q;QAFV,UAAU,c;QACV,  
WAAW,gB;QACD,2B;QAAV,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAL,W  
AAI,GAAL,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;IAoBA,uC;MAQI,UAAe,eAAL,S  
AAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MAC  
X,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;MAMI,IAAN,I;MACH,kBADs,SACT,c;QAAoB  
,4BAAc,SAAd,C;;QACZ,iCAAA,sBAAb,C;MAFZ,W;K;IAMJ,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,G  
AAI,EAAO,KAAp,C;MACJ,OAAO,G;K;8EAGX,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,  
mBAA1B,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAAC,UAAU,O  
AAV,CAAL,C;YAAyB,OAAO,K;;QACiD,OAAO,I;O;KARX,C;IAWA,2B;MAMI,IAAI,oCAAJ,C;QAAwB,OAA  
O,CAAC,mB;MACHC,OAAO,oBAAW,U;K;+EAGtB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAA  
sB,mBAA1B,C;UAAqC,OAAO,K;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAA  
V,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KARX,C;IAWA,6B;MAMoB,Q;MAFhB,IAAI,oCAAJ,C;QAA  
wB,OAAO,c;MAC/B,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,oBAAMb,qBAAnB,E  
AAmB,KAAAnB,E;;MACTB,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,yB;MAAA,gD;MAAA,wE;MAA



A,uC;QAMoB,Q;QAFhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,C;QAC5C,YAAY,C;QACI,2B;QAAhB,OA  
AgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAmB,qBAAnB,EAAmB,KAAhB,  
E;;QAC9C,OAAO,K;O;KAPX,C;gFAUA,yC;MAUoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAh  
B,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,wE;  
MAAA,gD;QAYoB,UAAiD,M;QAFjE,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAg  
B,yB;UAAM,cAAc,UAAU,oBAAmB,cAAhB,EAAmB,sBAAnB,UAAV,EAAuB,WAAvC,EAAoD,OAApD,C;;Q  
ACpC,OAAO,W;O;KAbX,C;OFAGBA,yC;MASI,kBAAkB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAA  
a,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAhB,EAA+B,WAA/B,C;;MAGtB,OA  
AO,W;K;wGAGX,yC;MAUI,kBAAkB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAAa,cAAb,C;QACf,O  
AAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAc,W  
AAtC,C;;MAGtB,OAAO,W;K;sFAGX,6B;MAKoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAA  
M,OAAO,OAAP,C;;K;OGAG1B,yB;MAAA,wE;MAAA,oC;QAOiB,UAAgC,M;QAD7C,YAAY,C;QACC,2B;QA  
Ab,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAhB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;O;K  
APvB,C;IAUA,0B;MAII,OAAO,sB;K;IAGX,2B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;kFAGX,+B;M  
AGW,sB;;QAUP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QA  
AS,O;QACvB,IAAI,CAAC,QAAS,UAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eAdmB,QAcJ,CAAS,OAAT,  
C;;UAEX,QAAQ,QAAS,O;UACjB,QAjBe,QAiBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,  
UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;MAvBP,yB;K;8FAGJ,+B;MAOI,eAA  
e,oB;MACf,IAAI,CAAC,QAAS,UAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS  
,UAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAA  
T,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,  
OAAO,O;K;mFAGX,yB;MAAA,sE;MF/yDA,iB;ME+yDA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C  
;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,  
QAAS,OAAIB,C;UACR,WFzzDG,MAAO,KEyzDO,QFzzDP,EEyzDiB,CFzzDjB,C;;QE2zDd,OAAO,Q;O;KApB  
X,C;mFAuBA,yB;MAAA,sE;MFj1DA,iB;MEi1DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAy  
B,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,  
OAAIB,C;UACR,WF31DG,MAAO,KE21DO,QF31DP,EE21DiB,CF31DjB,C;;QE61Dd,OAAO,Q;O;KApBX,C;m  
FAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,MAAM,6B;QAC  
/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,I  
AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFp3DA,iB;MEo  
3DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,OAAO,I;QACHc,eAAe,SAAS,QAAS,OA  
AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF53DG,MAAO,KE43DO,  
QF53DP,EE43DiB,CF53DjB,C;;QE83Dd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MFp5DA,iB;MEo5DA,sC;QAWI,e  
AAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,OAAO,I;QACHc,eAAe,SAAS,QAAS,OAAIB,C;QACf,OA  
AO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF55DG,MAAO,KE45DO,QF55DP,EE45Di  
B,CF55DjB,C;;QE85Dd,OAAO,Q;O;KAIBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAd,C  
;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,  
QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;OFAGX,yB;MA  
AA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS  
,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,S  
AAQ,QAAR,EAAkB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;sGAu  
BA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAI  
B,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,E  
AAkB,CAAIB,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MA  
Cf,IAAI,CAAC,QAAS,UAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;Q  
ACI,QAAQ,QAAS,O;QACjB,MFn+DG,MAAO,KEm+DE,GFn+DF,EE+DO,CFn+DP,C;;MEq+Dd,OAAO,G;K;  
IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MAC  
nB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFn+DG,MAAO,KE+/DE,GF//DF,EE+/DO,CF//DP,

C;;MEigEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc ,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAJ, C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,gD;MAKI,eAAe,oB ;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB ,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC, MAAM,C;;MAE9C,OAAO,G;K;IAGX,0B;MAIL,OAAO,sB;K;IAGX,2B;MAIL,OAAO,uB;K;IAGX,2B;MAGI,OA AO,uB;K;kFAGX,+B;MAGW,sB;;QAUP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP ,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eAdm B,QAcJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAjBe,QAiBP,CAAS,CAAT,C;UACR,IAAI,2BAAW, CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;MAvBP,yB;K; 8FAGJ,+B;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MAC vB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QA CjB,QAAQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,Q AAT,QAAS,W;MACIB,OAAO,O;K;mFAGX,yB;MAAA,sE;MF14DA,iB;MEk4DA,sC;QAaI,eAAe,oB;QACf,IAA I,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAh B,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF54DG,MAAO,KE44DO,QF54DP,EE44DiB,CF54DjB,C;;Q E84Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MFp6DA,iB;MEo6DA,sC;QAaI,eAAe,oB;QACf,IAAI,C AAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C ;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF96DG,MAAO,KE86DO,QF96DP,EE86DiB,CF96DjB,C;;QEg7 Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd ,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SA AS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;+ FAuBA,yB;MFv8DA,iB;MEu8DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QA ChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR ,WF/8DG,MAAO,KE+8DO,QF/8DP,EE+8DiB,CF/8DjB,C;;QEi9Dd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MFv+D A,iB;MEu+DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,Q AAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF/+DG,MAAO, KE++DO,QF/+DP,EE++DiB,CF/+DjB,C;;QEi/Dd,OAAO,Q;O;KAIBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IA AAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAh B,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,WAAW,C;;MAGnB,OA AO,Q;K;0FAGX,yB;MAAA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM ,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C ;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGnB,OAA O,Q;O;KApBX,C;SgAuBA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,e AAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI ,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;IAGX, gC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OA AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MfjtjEG,MAAO,KEsjEE,GfjtjEF,EEsjEO,CfjtjEP,C;;MEwj Ed,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU, QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MfllIEG,MAAO,KEklIEE,GfllIEF,EE klIEO,CfllIEP,C;;MEolEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OA AO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM, CAAN,KAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,gD;M AKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO, QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAA jC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,mB;MAC/B,OA AO,CAAC,oBAAW,U;K;iFAGvB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAASB,mBAA1B,C;UA AqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,

OAAO,K;;QACrD,OAAO,I;O;KARX,C;oFAWA,6B;MAKmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MArnBA,wE;MAqnBA,2BAQiB,yB;QA7nBjB,wE;eA6nBiB,0B;UAAA,4B;YAAE,aAAe,c;YAtnBjB,gB;YADb,YAAY,C;YACC,2B;YAAb,OAAa,cAAb,C;cAAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;YAsnBmB,W;W;S;OAAzB,C;MARjB,oC;QA9mBiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;QAsnBnB,gB;O;KARJ,C;oFAWA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAKbMD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuB,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;8GAuBA,yB;MAAA,wE;MAAA,uC;QAKbMD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHC,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuB,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;8GAuBA,gC;MACI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;QAEIB,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAcC,WAA1C,C;;QAEIB,OAAO,W;O;KArBX,C;wHAwBA,gC;MAaI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,YAAY,QAAS,gB;QACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAcC,WAA1C,C;;MAEIB,OAAO,W;K;0GAGX,gC;MACI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAJhB,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;QAARd,aHjJFO,W;QGkjFP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAmBoB,UACY,M;QAN5B,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;QAARd,aH1kFO,W;QG2kFP,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;kGA0BA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAcI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACHC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aHrmFO,W;QGsmFP,OAAO,QAAS,UAAhB,C;UACI,gBAAc,UAAU,aAAV,EAAuB,QAAS,OAAhC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KATBX,C;gHAyBA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAOBgC,Q;QAN5B,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACHC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aH9nFO,W;QG+nFP,YAAY,C;QACZ,OAAO,QAAS,UAAhB,C;UACI,gBAAc,WAAU,YAAV,EAAU,oBAAV,SAAmB,aAAnB,EAAgC,QAAS,OAAzC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;gFA0BA,yB;MArGA,kF;MAAA,gD;MAAA,gE;MAqGA,gD;QAcW,sB;;UAIGS,Q;UAJhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,qBAAO,OAgZ,OArGY,C;YAAP,uB;;UACqB,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBAoGIC,OApGkC,C;UAArD,aHjJFO,W;UGkjFP,kBAmGmB,O;UAIGH,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAiGwB,SAjGV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;QA8FP,yB;

O;KAdJ,C;8FAiBA,yB;MA9FA,kF;MAAA,gD;MAAA,gE;MA8FA,gD;QAeW,6B;;UA1FS,gB;UALhB,oBAAoB, mCAAwB,CAAxB,C;UACpB,IAAI,kBAaIB,CAArB,C;YAAwB,4BAAO,OA8FL,OA9FK,C;YAAP,8B;;UACqB, kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBA6F3B,OA7F2B,C;UAArD,aH1kFO,W;UG2kFP,YAAY,C ;UACZ,kBA2F0B,O;UA1FV,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAyF+B,SazFjB,EAAU,cAAV,E AAU,sBAAV,WAAmB,WAAhB,EAAGC,OAAhC,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAsFP, gC;O;KafJ,C;kfAkBA,+B;MAOoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QA CZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACF,2B;MA AhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q; MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,O AAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ, YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADh B,Y;QACgB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O ;KAbX,C;mFagBA,yB;MjB/7EA,6B;MiB+7EA,sC;QAWoB,Q;QADhB,UjB/7EmC,ciB+7EnB,CjB/7EmB,C;QiBg 8EnB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MjBnwFiD,ciBmwFjD,GjBnwF2D,KAAK,GiBmwFzD,S AAS,OAAT,CjBnwFoE,KAAX,IAAf,C;;QiBqwFrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MD78EA,+B;MC68EA,s C;QAWoB,Q;QADhB,UD58EqC,eAAW,oBC48E/B,CD58E+B,CAAX,C;QC68ErB,2B;QAAhB,OAAGB,cAAhB, C;UAAgB,yB;UACZ,MDjxFmD,eCixFnD,GDjxF8D,KAAK,KCixF5D,SAAS,OAAT,CDjxFuE,KAAX,CAAhB,C; ;QCmxFvD,OAAO,G;O;KAdX,C;IAiBA,qC;MAIoB,UAMT,M;MANS,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,y B;QACZ,IAAI,eAAJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAXB,MAAZB,C;;;MAId,OAAO,mE;K;IAGX,qC;M AIoB,UAMT,M;MANS,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,eAAJ,C;UACI,MAAM,gCAAyB ,2BAAwB,SAAXB,MAAZB,C;;;MAId,OAAO,+D;K;IAGX,kC;MAWI,OAAO,oBAAS,IAAT,EAAe,IAAf,EAAc, IAAT,C;K;IAGX,+C;MAGBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAT,C,EAAwD,SAAXD,C;K;IAGX,mC; MAIL,aAAa,iBAAa,mCAAwB,EAAXB,CAAb,C;MACb,kBAAc,KAAd,C;MAnIEgB,Q;MAAA,OAoIET,SApIES, W;MAAhB,OAAGB,cAAhB,C;QAAGB,2B;QAAU,oB;QAoIEK,IAAI,CAAC,SAAD,IAAY,OAplEX,SAoIEW,UA AhB,C;UAAiC,YAAU,I;UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QAplEvE,qB;UAoIED,MAplEqC,WAAI,SAA J,C;;MAoIED,OAAqB,M;K;IAGzB,sC;MAQI,IAAI,QpB0yJG,YAAQ,CoB1yJf,C;QAAwB,OAAY,SAAL,SAAK, C;MACpC,YAAqB,8BAAT,QAAS,C;MAtoEd,kBAAY,gB;MA4BH,Q;MAAA,OA2mET,SA3mES,W;MAAhB,O AAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CA2mEF,qBA3mEa,OA2mEb,CA3mEF,C;UAAyB,WAAy,WAAI,O AAJ,C;;MA2mE3D,OA1mEO,W;K;IA6mEX,sC;MAQI,YAAqB,gCAAT,QAAS,EAAGC,SAAhC,C;MACrB,IAAI, KAAM,UAAV,C;QACI,OAAY,SAAL,SAAK,C;MAppET,kBAAY,gB;MA4BH,Q;MAAA,OAynET,SAznES,W; MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAynEF,qBAznEa,OAynEb,CAznEF,C;UAAyB,WAAy,W AAI,OA AJ,C;;MAynE3D,OAxnEO,W;K;IA2nEX,sC;MAQI,YAAqB,8BAAT,QAAS,C;MACrB,IAAI,KAAM,UA AV,C;QACI,OAAY,SAAL,SAAK,C;MAIqET,kBAAY,gB;MA4BH,Q;MAAA,OAuoET,SAvoES,W;MAAhB,OA AgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAuoEF,qBAvoEa,OAuoEb,CAvoEF,C;UAAyB,WAAy,WAAI,OA AJ, C;;MAuoE3D,OAtoEO,W;K;8FAyoEX,yB;MAAA,8C;MAAA,qC;QAKI,OAAO,iBAAM,OAAN,C;O;KALX,C;0F AQA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACG,2B;QAAhB,O AAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OA AV,CAAJ,C;YACI,KAAM,WAAI,OA AJ,C;;YAEN,MAAO, WAAI,OA AJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;IAoBA,kC;MAIL,IAAI,oCAAJ,C;QA AwB,OAAY,OAAL,SAAK,EA AK,OAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACP,M AAO,WAAI,OA AJ,C;MACP,OAAO,M;K;IAGX,oC;MAIL,aAAa,iBAAa,iBAAO,CAAP,IAAb,C;MACb,MAAO,g BAAO,SAAP,C;MACP,MAAO,WAAI,OA AJ,C;MACP,OAAO,M;K;IAGX,qC;MAIL,IAAI,oCAAJ,C;QAAwB,O AAY,OAAL,SAAK,EA AK,QAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,M AAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAIL,aAAa,iBAAa,SAAK,KAAL,GAAY,QAAS,OAARb,IA Ab,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MA II,IAAI,oCAAJ,C;QAAwB,OAAY,OAAL,SAAK,EA AK,QAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAA O,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAIL,IAAI,mCAAJ,C;QACI,a AAa,iBAAa,SAAK,KAAL,GAAY,QAAS,KAARb,IAAb,C;QACb,MAAO,gBAAO,SAAP,C;QACP,MAAO,gBAA O,QAAP,C;QACP,OAAO,M;;QAEP,eAAa,iBAAa,SAAb,C;QACN,OAAP,QAAO,EAAO,QAAP,C;QACP,OAAO

,Q;;K;IAIf,qC;MAII,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MA  
CP,OAAO,M;K;IAGX,qC;MAII,aAAa,iBAaA,SAAK,KAAL,GAAY,EAAZ,IAAb,C;MACb,MAAO,gBAAO,SAA  
P,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;4FAGX,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,  
gBAAK,OAAL,C;O;KALX,C;8FAQA,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;IA  
QA,yD;MAgB+C,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MAOzE,Q;MANX,oBAAoB,IAApB,EAA0  
B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAAsB,  
WAAW,IAAX,KAAMb,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;QACrB,aAAa,iBAAMb,cAAAnB,C;QACb,g  
BAAY,CAAZ,C;QACA,Y;UAAO,c;UAAP,MAAgB,CAAT,mBAAiB,QAAxB,E;YAAA,K;UACI,iBAAsB,eAAL,  
IAAK,EAAa,WAAW,OAAX,IAAb,C;UACtB,IAAI,aAAa,IAAb,IAAqB,CAAC,cAA1B,C;YAA0C,K;Ud59FID,W  
AAW,iBc69Fa,Ud79Fb,C;UWCX,mBAAc,CAAd,YG49FwB,UH59FxB,Y;YXA6B,ec49FS,sBH39F3B,OG29FgC,  
GAAK,OAAL,IAAL,Cd59FT,C;;Uc49FrB,MAAO,Wd39FR,Ic29FQ,C;UACP,oBAAS,IAAT,I;;QAEJ,OAAO,M;;  
MAEX,eAAa,gB;MACiE,kBAA9E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAYC,cAAZC,EAAuE,K  
AAvE,C;ME5IGA,OAAGb,qBAAhB,C;QAAGb,gC;QF6lGL,mBE7lGqB,OF6lGrB,C;;MAEX,OAAO,Q;K;IAGX,s  
E;MAkBkD,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,oBAAoB,IAApB,EAA0B,IAA1B,C;MA  
CA,IAAI,0CAAwB,8BAA5B,C;QACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAAsB,WAAW,IAAX,K  
AAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;QACrB,aAAa,iBAaA,cAAb,C;QACb,eAAa,kBAAc,SAAd,C;  
QACb,YAAy,C;QACZ,OAAGb,CAAT,qBAAiB,QAAxB,C;UACI,iBAAsB,eAAL,IAAK,EAAa,WAAW,KAAX,I  
AAb,C;UACtB,IAAI,CAAC,cAAD,IAAmB,aAAa,IAApC,C;YAA0C,K;UAC1C,QAAO,cAAK,KAAL,EAAY,QA  
AQ,UAAR,IAAZ,C;UACP,MAAO,WAAI,UAAU,QAAY,CAAJ,C;UACP,gBAAS,IAAT,I;;QAEJ,OAAO,M;;MA  
EX,eAAa,gB;MACgE,kBAA7E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAYC,cAAZC,EAAuE,IAAv  
E,C;MEtoGA,OAAGb,qBAAhB,C;QAAGb,gC;QFuoGL,mBAAI,UEvoGiB,OFuoGjB,CAAJ,C;;MAEX,OAAO,Q;  
K;IAGX,kC;MAqBoB,gB;MAHhB,gBAXW,KAWW,O;MACtB,WAAW,iBF17FJ,MAAO,KE07FgB,mCAAwB,E  
AAxB,CF17FhB,EE07F6C,SF17F7C,CE07FH,C;MACX,QAAQ,C;MACQ,2B;MAAhB,OAAGb,cAAhB,C;QAAG  
B,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhBqB,GAgBP,OAHO,EAAnB,KAgBqB,CAA  
M,UAAN,EAAM,kBAAN,SAhBF,CAGBrB,C;;MAhBT,OAKBO,I;K;+EafX,yB;MAAA,kF;MAAA,gE;MFv7FA,i  
B;MEu7FA,8C;QAWoB,UAEsB,M;QALtC,gBAAGb,KAAM,O;QACtB,WAAW,eF17FJ,MAAO,KE07FgB,mCA  
AwB,EAAxB,CF17FhB,EE07F6C,SF17F7C,CE07FH,C;QACX,QAAQ,C;QACQ,2B;QAAhB,OAAGb,cAAhB,C;  
UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,OAAY,EAAMb,MAAM,UAA  
N,EAAM,kBAAN,SAAnB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkBA,kC;MAkBI,YAAy,oB;MACZ,aAZW,KA  
YQ,W;MACnB,WAAW,iBFv9FJ,MAAO,KEu9FgB,mCAAwB,EAAxB,CFv9FhB,EEu9FmD,wBAbtD,KAsD,EA  
AwB,EAAxB,CFv9FnD,CEu9FH,C;MACX,OAAO,KAAM,UAAN,IAAmB,MAAO,UAAjC,C;QACI,IAAK,Wafq  
B,GAeP,KAAM,OAfC,EAeO,MAAO,OAfd,CAerB,C;;MAfT,OAIBo,I;K;+EAdX,yB;MAAA,kF;MAAA,gE;MFv  
9FA,iB;MEm9FA,8C;QAQI,YAAy,oB;QACZ,aAAa,KAAM,W;QACnB,WAAW,eFv9FJ,MAAO,KEu9FgB,mCA  
AwB,EAAxB,CFv9FhB,EEu9FmD,wBAAN,KAAM,EAAwB,EAAxB,CFv9FnD,CEu9FH,C;QACX,OAAO,KAAM,  
UAAN,IAAmB,MAAO,UAAjC,C;UACI,IAAK,WAAI,UAAU,KAAM,OAAb,EAAwB,MAAO,OAAB,CAAJ  
,C;;QAET,OAAO,I;O;KAdX,C;IAiBA,gC;MASW,sB;;QAAP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAy  
B,qBAAO,W;UAAP,uB;;QACzB,ad/pGoD,gB;QcggGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UACI  
,WAAW,QAAS,O;UACpB,MAAO,WAnBkB,GAmBJ,OAnBI,EAmBK,IAnBL,CAMBlB,C;UACP,UAAU,I;;QAE  
d,qBAAO,M;;MAtBP,yB;K;8FAGJ,yB;MAAA,qD;MdZpGA,+D;McyPGa,uC;QAUI,eAAe,oB;QACf,IAAI,CAA  
C,QAAS,UAAAd,C;UAAyB,OAAO,W;QACChC,ad/pGoD,gB;QcggGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UA  
AhB,C;UACI,WAAW,QAAS,O;UACpB,MAAO,WAAI,UAAU,OAAY,EAAMb,IAAnB,CAAJ,C;UACP,UAAU,I;  
;QAEd,OAAO,M;O;KAnBX,C;IASBA,8F;MAQ6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;  
QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGtN  
,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACI,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QACZ  
,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;U  
ACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1  
B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4F;MAQwC  
,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MA

AI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC  
,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAATe,CAAI,F,W;K;4FAG5F,qB;MAKI,OA  
AO,S;K;IASS,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAN3B,iC;MAMI,oCAAgB,8BAAhB,C;K;IAGJ,+B;  
MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA  
AO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAASe,CAAAb,GAAGB,wCAAO,IAAvB,GA  
AgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAA  
hB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAASe,CAAAb,  
GAAGB,wCAAO,IAAvB,GAAGC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MA  
CD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;  
MAEJ,OAAW,UAASe,CAAAb,GAAGB,wCAAO,IAAvB,GAAGC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAA  
kB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,q  
BAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAASe,CAAAb,GAAGB,wCAAO,IAAvB,GAAGC,MAAM,K;K;IAGjD,  
+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,  
OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAASe,CAAAb,GAAGB,wCAAO,IAAvB,  
GAAGC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,c  
AAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAASe,CA  
Ab,GAAGB,wCAAO,IAAvB,GAAGC,MAAM,K;K;IAGjD,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,  
OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAe,C;M  
ACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;M  
ADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAAO,G;K;I  
AGX,2B;MAMoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,cAAO,OAAP,C;;M  
AEJ,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q  
ACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAh  
B,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IGn1GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,C  
AAN,EAAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,E  
AAW,CAAX,CAAX,IAA4B,CAAhC,GAAMC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;M  
ACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAAb,CAAX,GA  
A6B,CAAjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;IA+GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,  
CAAN,EAAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,  
EAAW,CAAX,CAAX,IAA4B,CAAhC,GAAMC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;  
MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAAb,CAAX,G  
AA6B,CAAjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;oGcNXX,yB;MAAA,iE;MAAA,uC;QASW,Q;QAAA,  
+B;;UAYS,U;UAAA,SjB4UoE,iBAAQ,W;UiB5U5F,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAbwB,SAAx,CAA  
U,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;UAGR,8BAAO,I;;QAIbA,kC;QAAA,iB;UAAmC,  
MAAM,gCAAuB,4DAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MAAA,OAAA,SjB4UoE,Q  
AAQ,W;MiB5U5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OA  
AO,M;;MAGf,OAAO,I;K;IAGX,6B;MAII,IAAI,mBAAQ,CAAZ,C;QACI,OAAO,W;MACX,eAAe,iBAAQ,W;M  
ACvB,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,W;MACX,YAAAY,QAAS,O;MACrB,IAAI,CAAC,QAAS,UAA  
Ad,C;QACI,OAAO,OjB8PiD,SiB9PiC,KjB8P+C,IAAL,EiB9PiC,KjB8PoD,MAAV,CiB9PiD,C;;MACX,aAAa,iBA  
AsB,cAAAtB,C;MACb,MAAO,WjB4PqD,SiB5PjD,KjB4PsD,IAAL,EiB5PjD,KjB4P2D,MAAV,CiB5PrD,C;;QAEw  
B,kBAAhB,QAAS,O;QAAPB,MAAO,WjB0PiD,SAAK,eAAL,EAAU,iBAAV,CiB1PiD,C;;MACO,QAAT,QAAS,  
W;MACIB,OAAO,M;K;uFAGX,yB;MAAA,+D;MASBA,gD;MatBA,uC;QAMW,kBAAU,gB;QAoBD,Q;QAAA,  
OjBqRoE,iBAAQ,W;QiBrR5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArB6B,SAqBIB,CAAU,OAAV,C;UACC,  
OAAZ,WAAAY,EAAO,IAAP,C;;QAtBhB,OAwbO,W;O;KA9BX,C;uFASA,yB;MAAA,+D;MAwBA,gD;MAxBA,  
uC;QAUW,kBAAU,gB;QASBD,Q;QAAA,OjBsQoE,iBAAQ,W;QiBtQ5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,  
WAvB6B,SAuBIB,CAAU,OAAV,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAxBhB,OA0BO,W;O;KApCX,C;2F  
AaA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,OAAA,SjBqRoE,QAAQ,W;QiBrR5F,OAAgB,cAAhB,C;UAAg  
B,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;2F

AWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,OAAA,SjBsQoE,QAAQ,W;QiBtQ5F,OAAgB,cAAhB,C;UA  
AgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAA,Y,EA AO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;  
8EAeA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,cAAb,C;QA2BA,Q;QAAA,OjB6NuE,iBAAQ,W;QiB7  
N5F,OAAa,cAAb,C;UAAa,sB;UACT,WAA,Y,WA5BiB,SA4Bb,CAAU,IAAV,CAAJ,C;;QA5BhB,OA6BO,W;O;K  
ApCX,C;4FAUA,yB;MAAA,+D;MAAA,uC;QAOW,kBAAa,gB;QA4EJ,Q;QAAA,OjBkKoE,iBAAQ,W;QiBIK5F,  
OAAgB,cAAhB,C;UAAgB,yB;UApEK,U;UAAA,cARe,SAQf,CAoEQ,OApER,W;YAA,sC,6B;;;QAR3D,OASO,W  
;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAYeOB,Q;QAAA,OjBkKoE,iBAAQ,W;QiBIK5F,OAAgB,cAAhB,C;UAA  
gB,yB;UApEK,U;UAAA,wBAoEQ,OApER,W;YAA,sC,6B;;;QAC3D,OAAO,W;O;KANX,C;kFASA,6C;MAKiB,  
Q;MAAA,OAAA,SjB6NuE,QAAQ,W;MiB7N5F,OAAa,cAAb,C;QAAa,sB;QACT,WAA,Y,WAAI,UAAU,IAAV,C  
AAJ,C;;MACHB,OAAO,W;K;8EAGX,gC;MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAAe,OAAO,I;MACN,OAAA,Sj  
BiNoE,QAAQ,W;MiBjN5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB  
,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAMI,OAAO,CAAC,mB;K;+EAGZ,gC;MAOoB,Q;MADhB,IAAI,mB  
AAJ,C;QAAe,OAAO,K;MACN,OAAA,SjB6LoE,QAAQ,W;MiB7L5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IA  
AI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,gC;  
MAMoB,Q;MAFhB,IAAI,mBAAJ,C;QAAe,OAAO,C;MACtB,YAA,Y,C;MACI,OAAA,SjB2KoE,QAAQ,W;MiB3  
K5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;sFA  
GX,6B;MAKoB,Q;MAAA,OAAA,SjBkKoE,QAAQ,W;MiBIK5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,  
OAAp,C;;K;kFAGIb,+B;MAemB,kBAAR,iB;MAAQ,sB;;QJkoDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;  
UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;  
UAAP,uB;;QACzB,eIjpDmB,QJipDJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIppDe,QJopDP,CAAS,C  
AAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QAC  
IB,qBAAO,O;;MI1pDP,yB;K;8FAGJ,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJkoDf,eAAe,sB;QACf,IAAI,CAAC,  
QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UA  
AyB,qBAAO,O;UAAP,uB;;QACzB,eItoD2B,QJsoDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIzoDuB  
,QJyoDf,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAA  
T,QAAS,W;QACIB,qBAAO,O;;MI/oDP,yB;K;mFAGJ,yB;MJ+oDA,sE;MF/yDA,iB;MMgKA,sC;QJ4pDI,eI/oDO,  
iBJ+oDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIjpDqB,QJipDN,CAAS,QAAS,OA  
AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QImpDiB,QJmpDT,CAAS,QAAS,OAAIB,C;UACR,WFzzDG,MAA  
O,KEyzDO,QFzzDP,EEyzDiB,CFzzDjB,C;;QMqKd,OJspDO,Q;O;KInqDX,C;mFAGBA,yB;MJspDA,sE;MFj1DA,  
iB;MM2LA,sC;QJmqDI,eItpDO,iBJspDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI  
pDqB,QJwpDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI1pDiB,QJ0pDT,CAAS,QAAS,  
OAAIB,C;UACR,WF31DG,MAAO,KE21DO,QF31DP,EE21DiB,CF31DjB,C;;QMgMd,OJ6pDO,Q;O;KI1qDX,C;  
mFAGBA,yB;MJ6pDA,sE;MI7pDA,sC;QJwqDI,eI7pDO,iBJ6pDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB  
,MAAM,6B;QAC/B,eI/pDqB,QJ+pDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIjqDiB,Q  
JiqDT,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QInqDnB,OJsqDO,Q;O;  
KIjrDX,C;+FACa,yB;MN9MA,iB;MM8MA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJsqDf,eAAe,sB;UACf,IAAI,C  
AAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIxD2B,QJwqDZ,CAAS,QAAS,OAAIB,C;UACf,O  
AAO,QAAS,UAAhB,C;YACI,QI1qDuB,QJ0qDf,CAAS,QAAS,OAAIB,C;YACR,WF53DG,MAAO,KE43DO,QF5  
3DP,EE43DiB,CF53DjB,C;;UE83Dd,qBAAO,Q;;QI7qDP,yB;O;KAXJ,C;+FACa,yB;MNvOA,iB;MMuOA,sC;QA  
WmB,kBAAR,iB;QAAQ,sB;;UJ6qDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;  
;UACzB,eI/qD2B,QJ+qDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIjrDuB,QJirDf,CAA  
S,QAAS,OAAIB,C;YACR,WF55DG,MAAO,KE45DO,QF55DP,EE45DiB,CF55DjB,C;;UE85Dd,qBAAO,Q;;QIpr  
DP,yB;O;KAXJ,C;+FACa,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJorDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAA  
d,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eItrD2B,QJsrDZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAh  
B,C;UACI,QIxrDuB,QJwrDf,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;Q  
AGnB,qBAAO,Q;;MI7rDP,yB;K;0FAGJ,yB;MJ6rDA,sE;MI7rDA,kD;QJwsDI,eI7rDO,iBJ6rDQ,W;QACf,IAAI,C  
AAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI/rDqC,QJ+rDtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QA  
AS,UAAhB,C;UACI,QIjsDiC,QJisDzB,CAAS,QAAS,OAAIB,C;UACR,IIIsDqB,UJksDN,SAAQ,QAAR,EAakB,C

AAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QInsDnB,OJssDO,Q;O;KIjtDX,C;sGAcA,2C;MASmB,kBAA  
R,iB;MAAQ,0B;;QJssDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eIx  
sD2C,QJwsD5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIIsDuC,QJ0sD/B,CAAS,QAAS  
,OAAIB,C;UACR,II3sD2B,UJ2sDZ,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;  
QAGnB,yBAAO,Q;;;MI/sDP,6B;K;sFAGJ,yB;MAOA,8D;MAPA,wC;QAIL,OASe,cAAR,iBAAQ,EATM,UASN,C  
;O;KAbnB,C;kGAOA,yB;MAAA,8D;MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAac,UAAAd,C;O;KANnB,C;kFA  
SA,+B;MAcmB,kBAAR,iB;MAAQ,sB;;QJwxDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;  
UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB  
,eIvyDmB,QJuyDJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QI1yDe,QJ0yDP,CAAS,CAAT,C;UACR,IA  
AI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;;M  
IhzDP,yB;K;8FAGJ,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJwxDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;U  
AAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;U  
AAP,uB;;QACzB,eI5xD2B,QJ4xDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QI/xDuB,QJ+xDf,CAAS,C  
AAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAC  
IB,qBAAO,O;;;MIryDP,yB;K;mFAGJ,yB;MJqyDA,sE;MF14DA,iB;MM6FA,sC;QJkzDI,eIryDO,iBJqyDQ,W;QAC  
f,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIvyDqB,QJuyDN,CAAS,QAAS,OAAIB,C;QACf,OA  
AO,QAAS,UAAhB,C;UACI,QIzyDiB,QJyyDT,CAAS,QAAS,OAAIB,C;UACR,WF54DG,MAAO,KE44DO,QF54  
DP,EE44DiB,CF54DjB,C;;QMkGd,OJ4yDO,Q;O;KIzDX,C;mFAGBA,yB;MJ4yDA,sE;MFp6DA,iB;MMwHA,sC;  
QJyzDI,eI5yDO,iBJ4yDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI9yDqB,QJ8yDN,  
CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIhzDiB,QJgzDT,CAAS,QAAS,OAAIB,C;UACR  
,WF96DG,MAAO,KE86DO,QF96DP,EE86DiB,CF96DjB,C;;QM6Hd,OJmzDO,Q;O;KIh0DX,C;mFAGBA,yB;MJ  
mzDA,sE;MInzDA,sC;QJ8zDI,eInzDO,iBJmzDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QA  
C/B,eIrdqB,QJqzDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIvzDiB,QJuzDT,CAAS,Q  
AAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QIzzDnB,OJ4zDO,Q;O;KIv0DX,C;+FA  
cA,yB;MN3IA,iB;MM2IA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJ4zDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAA  
d,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eI9zD2B,QJ8zDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAA  
hB,C;YACI,QIh0DuB,QJg0Df,CAAS,QAAS,OAAIB,C;YACR,WF/8DG,MAAO,KE+8DO,QF/8DP,EE+8DiB,CF/  
8DjB,C;;UEi9Dd,qBAAO,Q;;;QIn0DP,yB;O;KAXJ,C;+FAcA,yB;MNpKA,iB;MMoKA,sC;QAWmB,kBAAR,iB;Q  
AAQ,sB;;UJm0Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eI9D2B,  
QJq0DZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIv0DuB,QJu0Df,CAAS,QAAS,OAAIB,  
C;YACR,WF/+DG,MAAO,KE++DO,QF/+DP,EE++DiB,CF/+DjB,C;;UEi/Dd,qBAAO,Q;;;QI10DP,yB;O;KAXJ,C  
;+FAcA,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJ00Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBA  
AO,I;UAAP,uB;;QACzB,eI50D2B,QJ40DZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI90  
DuB,QJ80Df,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,  
Q;;;MIIn1DP,yB;K;0FAGJ,yB;MJm1DA,sE;MIIn1DA,kD;QJ81DI,eIn1DO,iBJm1DQ,W;QACf,IAAI,CAAC,QAAS,  
UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI91DqC,QJq1DtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C  
;UACI,QIv1DiC,QJu1DzB,CAAS,QAAS,OAAIB,C;UACR,IIx1DqB,UJw1DN,SAAQ,QAAR,EAakB,CAAIB,CA  
AX,GAakC,CAAtC,C;YACI,WAAW,C;;;QIz1DnB,OJ41DO,Q;O;KIv2DX,C;sGAcA,2C;MASmB,kBAAR,iB;M  
AAQ,0B;;QJ41Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eI91D2C,  
QJ81D5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIh2DuC,QJg2D/B,CAAS,QAAS,OAA  
IB,C;UACR,IIj2D2B,UJi2DZ,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGn  
B,yBAAO,Q;;;MIr2DP,6B;K;IAGJ,0C;MAGI,OASe,gBAAR,iBAAQ,EATM,UASN,C;K;kGANnB,yB;MAAA,8D;  
MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAac,UAAAd,C;O;KANnB,C;IASA,4B;MAMI,OAAO,mB;K;iFAGX,gC;  
MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAAe,OAAO,I;MACN,OAAA,SjBnJoE,QAAQ,W;MiBmJ5F,OAAGB,cAAh  
B,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;oFAGX,6B;MAK  
mC,Q;MAAA,OjB5JqD,iBAAQ,W;MiB4J7E,OAAGB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,g  
B;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MJwyCA,wE;MIxyCA,2BAQIB,yB;QJgyCjB,wE;eIhyCiB,0B;UAAA,4B;  
YAAU,kBAAR,iB;YAAQ,aAAe,c;YJuyCzB,gB;YADb,YAAY,C;YACC,6B;YAAb,OAAa,cAAb,C;cAAa,sB;cAA



M,OAAO,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;YIvyC2B,W;W;S;OAAjC,C;MARjB,oC ;QJ+yCiB,gB;QADb,YAAy,C;QACC,OIvyCE,iBJuyCF,W;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBA AmB,cAAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;QIvyCnB,gB;O;KARJ,C;4FAWA,qB;MAKI,OAAO,iB ;K;IAGX,iC;MAII,OAAe,aAAR,iBAAQ,C;K;IC9hBnB,kC;MAEI,gBCmE2D,8BAAy,c;MDIEvE,IAAI,SAAU,OA AV,GAAMb,CAAvB,C;QACW,Q;QAAA,IAAI,cAAQ,GAAZ,C;UAAA,OAAsB,S;;uBAAe,qBAAU,CAAV,C;UA AA,YAAe,SEiNc,WFjNM,CEiNN,Caff,c;UFIMnD,OG8MoD,2BAAL,GAakB,K;;QH9MxE,W;;MAEJ,OAAuB,o BAAhB,wBAAgB,C;K;gFxBD3B,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KAR X,C;gFAWA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MA AA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;;QAQQ,OAAc,QAAP,MA AO,EAAQ,SAAR,C;;QACbB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OA Ac,SAAP,MAAO,EAAS,SAAT,C;;QACbB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC ;QAQQ,OAAiD,OAA1C,MAAO,iBAAQ,e4BtCgB,I5BsCxB,EAAoB,CAAA,c4BtCI,I5BsCJ,IAAY,CAAZ,IAAp B,CAAmC,C;;QACnD,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAA A,uD;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QA OI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAQI,OAAO,wBAAa,cA Ab,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,QAAP,MAAO,EAAQ,SAAR, C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,SAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C ;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAiD,OAA1C,MAAO,iBAAQ,e4BxGoB,I5BwG5B,EAAoB,C AAA,c4BxGQ,I5BwGR,IAAY,CAAZ,IAApB,CAAmC,C;K;mFAGrD,8B;MAQI,OAAO,mBAAMb,2BAAS,OAA T,C;K;oFAG9B,8B;MAQI,OAAO,mBAAMb,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OAAO,mBAAMb,2BAAS, OAAT,C;K;IAG9B,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf, C;K;IAGX,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,O AAO,2BAAe,KAAf,C;K;IAGX,uC;MgBzHW,ShBgIM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAA S,EAAT,CAAhB,GAakC,K;K;IAG5E,uC;MgBnIW,ShB0IM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2 BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgB7IW,ShBoJM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GA AgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgBvJW,ShB8JM,qBAAN,KAAM,C;MAAb,OAA4C,UAA J,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG9E,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MgBzKW ,ShBgLM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG5E,uC;M gBnLW,ShB0LM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E ,uC;MgB7LW,ShBoMM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K; IAG7E,uC;MgBvMW,ShB8MM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAA kC,K;K;IAG9E,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C; K;IAGX,uC;MgBjOW,ShBsOM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAA kC,K;K;IAG7E,uC;MgBzOW,ShB8OM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAh B,GAakC,K;K;IAG9E,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,u C;MgBrQW,ShB0QM,iBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IA G5E,uC;MgB7QW,ShBkRM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC, K;K;IAG7E,uC;MgBrRW,ShB0RM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,G AakC,K;K;IAG9E,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAS,KAAM,WAA f,C;K;IAGX,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;I AGX,uC;MgBjUW,ShBsUM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC, K;K;IAG7E,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,+B;MAOI,O AAO,sCAAe,yBAAgB,SAAhB,EAAYB,EAzB,EAakC,EAaIC,C;K;IAG1B,iC;MAOI,OAAO,uCAAqB,yBAAg B,SAAhB,EAAYB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAj C,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;I AG1B,iC;MAOI,OAAO,uCAAqB,yBAAgB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,iC;MAOI,OAA O,sCAAe,yBAAgB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG1B,iC;MAOI,OAAO,uCAAqB,yBAAgB,S AAhB,EAAYB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAA8B,EAA9B,



B, YAAzB, GAA2C, S; K; IAGtD, kD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, kD; MAQI, OAAW, 0BAAO, YAAP, KAAJ, GAAyB, YAAzB, GAA2C, S; K; IAGtD, kD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, kD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, +C; MAQI, OAAW, 4BAAO, YAAP, KAAJ, GAAyB, YAAzB, GAA2C, S; K; IAGtD, iD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, iD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, iD; MAQI, OAAW, 0BAAO, YAAP, KAAJ, GAAyB, YAAzB, GAA2C, S; K; IAGtD, iD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, iD; MAQI, OAAW, YAAO, YAAX, GAAyB, YAAzB, GAA2C, S; K; IAGtD, yD; MAQI, IAAI, iBAAiB, IAAjB, IAAYB, iBAAiB, IAA9C, C; QACI, IAAI, +BAAe, YAAf, KAAJ, C; UAAiC, MAAM, gCAAYB, 6DAAiD, YAAjD, wCAAoF, YAApF, OAAzB, C; QACvC, IAAI, 4BAAO, YAAP, KAAJ, C; UAAyB, OAAO, Y; QAChC, IAAI, 4BAAO, YAAP, KAAJ, C; UAAyB, OAAO, Y; QAGhC, IAAI, iBAAiB, IAAjB, IAAYB, 4BAAO, YAAP, KAA7B, C; UAAkD, OAAO, Y; QACzD, IAAI, iBAAiB, IAAjB, IAAYB, 4BAAO, YAAP, KAA7B, C; UAAkD, OAAO, Y; MAE7D, OAAO, S; K; IAGX, 2D; MAQI, IAAI, eAAe, YAAAnB, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, 2D; MAQI, IAAI, eAAe, YAAAnB, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, 2D; MAQI, IAAI, eAAe, YAAAnB, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, 2D; MAQI, IAAI, 6BAAe, YAAf, KAAJ, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, yCAAoF, YAApF, iBAAzB, C; MACvC, IAAI, 0BAAO, YAAP, KAAJ, C; QAAyB, OAAO, Y; MACHc, IAAI, 0BAAO, YAAP, KAAJ, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, 2D; MAQI, IAAI, eAAe, YAAAnB, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, 2D; MAQI, IAAI, eAAe, YAAAnB, C; QAAiC, MAAM, gCAAYB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, IAAI, YAAO, YAAX, C; QAAyB, OAAO, Y; MACHc, OAAO, S; K; IAGX, sC; MAUW, Q; MADP, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAYB, 4CAAyC, KAAzC, MAAzB, C; MAGvB, IAAA, KAAM, 0BAAiB, SAAjB, EAAuB, KAAM, MAA7B, CAAN, IAA6C, CAAC, KAAM, 0BAAiB, KAAM, MAAvB, EAA8B, SAA9B, CAAPd, C; QAAiG, OAAO, KAAM, M; WAEjG, IAAA, KAAM, 0BAAiB, KAAM, aAAvB, EAAqC, SAARc, CAAN, IAAoD, CAAC, KAAM, 0BAAiB, SAAjB, EAAuB, KAAM, aAA7B, CAA3D, C; QAA+G, OAAO, KAAM, a; QACvG, gB; MALZ, W; K; IASJ, sC; MAYW, Q; MAJP, IAAI, 8CAAJ, C; QACI, OAAO, WAAL, SAAK, EAAy, KAAZ, C; MAEhB, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAYB, 4CAAyC, KAAzC, MAAzB, C; MAEvB, gCAAO, KAAM, MAAb, M; QAA4B, OAAO, KAAM, M; WAC5B, gCAAO, KAAM, aAAb, M; QAAmC, OAAO, KAAM, a; QAC3B, gB; MAHZ, W; K; IAOJ, sC; MAYW, Q; MAJP, IAAI, 8CAAJ, C; QACI, OAAO, WAAL, SAAK, EAAc, KAAAd, C; MAEhB, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAYB, 4CAAyC, KAAzC, MAAzB, C; MAEvB, gBAAO, KAAM, MAAb, C; QAA4B, OAAO, KAAM, M; WAC5B, gBAAO, KAAM, aAAb, C; QAAmC, OAAO, KAAM, a; QAC3B, gB; MAHZ, W; K; IAOJ, sC; MAYW, Q; MAJP, IAAI, 8CAAJ, C; QACI, OAAO, WAAL, SAAK, EAAe, KAAf, C; MAEhB, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAYB, 4CAAyC, KAAzC, MAAzB, C; MAEvB, 8BAAO, KAAM, MAAb, M; QAA4B, OAAO, KAAM, M; WAC5B, 8BAAO, KAAM, aAAb, M; QAAmC, OAAO, KAAM, a; QAC3B, gB; MAHZ, W; K; IW1rCJ, oD; MAMuF, wC; K; IANvF, 8CAOI, Y; MAAuC, 8B; K; IAP3C, gF; IkbQA, yC; MAMI, OAAO, sBAAQ, OAAO, KAAoB, C; K; IAWG, 2C; MAAA, qB; QAAE, MAAM, 8BAA0B, +CAA4C, aAA5C, MAA1B, C; O; K; IAR1C, uC; MAQI, OAAO, 8BAAgB, KAAhB, EAAuB, yBAAvB, C; K; IAGX, 4D; MACqB, Q; MANjB, IAAI, QAAQ, CAAZ, C; QACI, OAAO, aAAa, KAAb, C; MACX, eAAe, oB; MACf, YAAy, C; MACZ, OAAO, QAAS, UAAhB, C; QACI, cAAc, QAAS, O; QACvB, IAAI, WAAS, YAAT, EAAS, oBAAT, OAAJ, C; UACI, OAAO, O; MAEf, OAAO, aAAa, KAAb, C; K; IAGX, 8C; MACqB, Q; MANjB, IAAI, QAAQ, CAAZ, C; QACI, OAAO, I; M ACX, eAAe, oB; MACf, YAAy, C; MACZ, OAAO, QAAS, UAAhB, C; QACI, cAAc, QAAS, O; QACvB, IAAI, WAAS, YAAT, EAAS, oBAAT, OAAJ, C; UACI, OAAO, O; MAEf, OAAO, I; K; 8EAGX, gC; MASW, sB; QA2FS, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAgB, yB; UAAM, IA3FH, SA2FO, CAAU, OAAV, CAAJ, C; YAAwB, qBAAO, O; YAAP, uB; QAC9C, qBAAO, I; MA5FP, yB; K; uFAGJ, gC; MAkOoB, Q; MADhB, WAAe, I; MACC, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, IA1Nc, SA0NV, CAAU, OAAV, CAAJ, C; UACI, OAAO, O; MA3Nf, OA8NO, I; K; IA3NX,

6B;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAUb,oBAAvB,C;MACV,OAAO,QAAS,O;K;iFAGpB,yB;MAAA,iE;MAAA,uC;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,sDAAvB,C;O;KARV,C;kGAWA,yB;MAAA,iE;MAAA,uC;QAWW,Q;QAAA,+B;;UAcS,U;UAAA,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAfwB,SAeX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QApBA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,iEAAvB,C;;QAAhD,OAAO,I;O;KAXX,C;8GAcA,gC;MAWob,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,mC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,OAAO,QAAS,O;K;6FAGpB,gC;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;IAGX,wC;MAOiB,Q;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAhB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;+FAGX,yB;MAAA,wE;MAAA,uC;QAOiB,Q;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAhB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAEJ,OAAO,E;O;KAbX,C;6FAGBA,yB;MAAA,wE;MAAA,uC;QAQiB,Q;QAFb,gBAAGB,E;QACHB,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAhB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAY,K;UACHB,qB;;QAEJ,OAAO,S;O;KAdX,C;IAiBA,4B;MAUI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAUb,oBAAvB,C;MACV,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;+EAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAYoB,UAQTM;QAVP,WAAe,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA,Y,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,2E;O;KApBX,C;IAuBA,4C;MAQiB,Q;MAFb,gBAAGB,E;MACHB,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAhB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAY,K;QACHB,qB;;MAEJ,OAAO,S;K;IAGX,kC;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;2FAGX,gC;MASoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO,O;;MAGf,OAAO,I;K;IAGX,8B;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAUb,oBAAvB,C;MACV,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,MAAM,gCAAyB,qCAAzB,C;MACV,OAAO,M;K;mFAGX,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAQoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYB,mDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA,Y,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,6E;O;KAjBX,C;IAoBA,oC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,OAAO,I;MACX,OAAO,M;K;+FAGX,gC;MAQoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAWW,Q;MhBhXP,IAAI,EgB+WI,KAAK,ChB/WT,CAAJ,C;QACI,cgB8Wc,sD;QhB7Wd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MgB+WN,UAAK,CAAL,C;QAAU,gB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAIb,EAAwB,SAAxB,C;K;IAGX,wC;MAQI,OAAO,sBAAkB,SAIb,EAAwB,IAAxB,EAA8B,SAA9B,C;K;IacqE,iD;MAAA,qB;QAAE,yBAAU,EAAG,MAAb,EAAoB,EAAG,MAAvB,C;O;K;IAAkC,oC;MAAE,OAAA,EAAG,M;K;IAXzH,+C;MAWI,OAAO,yBAAqB,sBAAkB,qBAAiB,SAAjB,CAAIB,EAA0C,IAA1C,EAAGD,+BAAhD,CAArB,EAAYG,sBAAzG,C;K;oGAGX,yB;MA80BA,wE;MA90BA,oD;QAu1BiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA50BT,IAAI,UA40BkB,oBAAmB,cAAhB,EAAmB,sBAAnB,UA50BIB,EA40B+C,IA50B/C,CAAJ,C;YAA2C,sBA40BQ,IA50BR,C;;QAE/C,OAAO,W;O;KAbX,C;sGAgBA,yB;MAAA,8C;MAAA,0C;MAAA,8B;MASkB,qD;QAAA,qB;UAAE,c;S;O;MATpB,sC;QASW,Q;QAAP,OAAO,uCAAo,iCAAP,gC;O;KATX,C;0GAYA,4C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAA,Y,WAAI,OAAJ,C;;MACpD,OAAO,W;K;IAGX,2C;MAQI,OAAO,sBAAkB,SAIb,EAAwB,KAAxB,EAA+B,SAAB,C;K;IAYU,kC;MAAE,iB;K;IATvB,oC;MASW,Q;MAAP,OAAO,4CAAU,oBAAV,kC;K;IAGX,mD;MAQoB,Q;MA

AA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAY,WAAI,OAAJ,C;;MACvD,  
OAAO,W;K;4FAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UA  
AU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;sFAGX,6C;MAQoB,Q;MAAA,2B;  
MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;  
MAC1D,OAAO,W;K;IAGX,8B;MAWW,Q;MhBzgBP,IAAI,EgBwgBI,KAAK,ChBxgBT,CAAJ,C;QACI,cgBugBc  
,sD;QhBtgBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MgBwgBN,UAAK,CAAL,C;QAAU,sB;WACV,+C;QAAiC,O  
AAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAMb,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sB  
AAkB,SAAlB,EAAwB,SAAxB,C;K;IAWA,2C;MAAA,8B;K;8CACH,Y;MACI,iBAA6B,iBAAZ,gBAAy,C;MAC  
IB,QAAX,UAAW,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6B;MAQI,0C;K;sFASJ,yB;MAAA,sD;MdjFA,sC;MAA  
A,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA  
5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;Mc0ef,sC;QAUI,OAAO,sBdpfP,eAAW,iBcofiB,  
QdpfjB,CAAX,CcofO,C;O;KAVX,C;0GAaA,yB;MAAA,sD;Md3eA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;  
eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,  
CAA3B,C;W;S;OA+EI,C;Mcoef,sC;QAQI,OAAO,sBd5eP,eAAW,2Bc4e2B,Qd5e3B,CAAX,Cc4eO,C;O;KARX,C;  
IAWA,uC;MAQI,OAAO,wBAAW,cAAAX,C;K;IAWA,uE;MAAA,sC;MAAA,4C;K;kDACH,Y;MACI,iBAAiC,iBA  
AhB,oBAAGB,C;MACtB,WAAX,UAAW,EAAS,uBAAT,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6C;MAQI,0D;K;  
wFASJ,yB;MAAA,wE;MAAA,uC;QAaW,kBAAy,oB;QAI FH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,y  
B;UACZ,WAI F sC,SAkFvB,CAAU,OAAV,C;UvBnEnB,wBAAl,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QuBfA,  
OAoFO,W;O;KAjGX,C;6FAGBA,yB;MAAA,wE;MAAA,yC;QAaW,kBAAc,oB;QA8BL,Q;QAAA,2B;QAAhB,O  
AAgB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA/B4B,WA+BxB,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA/Bh  
B,OAI CO,W;O;KA9CX,C;6FAGBA,yB;MAAA,wE;MAAA,yD;QAYW,kBAAc,oB;QAI CL,Q;QAAA,2B;QAAhB,  
OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aAIC4B,WakCxB,CAAY,OAAZ,CAAJ,EAlCyC,cAkCf,CAAe,OA  
Af,CAA1B,C;;QAICHB,OAoCO,W;O;KAhDX,C;iGAeA,+C;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;Q  
AAgB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEHb,OAAO,W;K;iGAGX,+D;MAY  
oB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eA  
Ae,OAaf,CAA1B,C;;MAEHb,OAAO,W;K;4FAGX,6C;MAWob,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAA  
gB,yB;QACZ,WAAe,UAAU,OAAV,C;QvBnEnB,wBAAl,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;MuBqEA,OA  
AO,W;K;gGAGX,yB;MAAA,wE;MAAA,2C;QACI,aAAa,oB;QAGBG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U  
AAgB,yB;UafO,MAgBP,aAAI,OAAJ,EAhBe,aAgBF,CAAc,OAAd,CAAb,C;;QAhBhB,OAAuB,M;O;Kaf3B,C;o  
GakBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cA  
Ac,OAAd,CAAb,C;;MAEHb,OAAO,W;K;IAGX,gD;MAMiB,Q;MAAA,2B;MAAb,OAaa,cAAb,C;QAAa,sB;QA  
CT,WAAY,WAAI,IAAJ,C;;MAEHb,OAAO,W;K;IAGX,gC;MAMI,OAAO,0BAAa,cAAb,C;K;IAGX,8B;MAMI,O  
AA4B,qBAAhB,iBAAL,SAAK,CAAGB,C;K;IAGhC,qC;MAMI,OAAO,0BAAa,gBAAb,C;K;IAGX,4B;MAQI,OA  
AwC,oBAAjC,0BAAa,sBAAb,CAAI,C;K;IAG5C,0C;MAYI,OAAO,uBAAMb,SAAnB,EAAYB,SAAZB,6BAAo  
C,qB;;OAApC,E;K;IAGX,0C;MAQI,OAAO,uBAAMb,SAAnB,EAAYB,SAAZB,6BAAoC,qB;;OAApC,E;K;IAGX  
,iD;MAaI,OAAO,kBAAE,SAAf,EAAqB,SAARb,6BAAgC,qB;;OAAhC,E;K;IAGX,iD;MAaI,OAAO,kBAAE,SAAf  
,EAAqB,SAARb,6BAAgC,qB;;OAAhC,E;K;sGAGX,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;Q  
AF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAMb,cAAnB,EA  
mB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEHb,OAAO,W;O;KAjBX,C;u  
GAoBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,  
cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAMb,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;U  
ACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEHb,OAAO,W;O;KAjBX,C;yFAoBA,yB;MAAA,gD;MAAA,oD;QUo  
B,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,E  
AAO,IAAP,C;;QAEHb,OAAO,W;O;KAdX,C;yFAiBA,yB;MAAA,gD;MAAA,oD;QAMoB,Q;QAAA,2B;QAAhB,  
OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEHb,  
OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,wE;MA6BA,+D;MA7BA,yC;QAWW,kBAAU,oB;QA6BD,Q;QAAA,2  
B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA9BiD,WA8BvC,CAAY,OAAZ,C;UvBjoBP,U;UADP,YuBm  
oBe,WvBnoBH,WuBmoBwB,GvBnoBxB,C;UACL,IAAI,aAAJ,C;YACH,auBioBuC,gB;YAA5B,WvBhoBX,auBg

oBgC,GvBhoBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuB6nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAhCT,Oak  
CO,W;O;KA7CX,C;qFAcA,yB;MAAA,wE;MAKCA,+D;MAICA,yD;QAYW,kBAAU,oB;QAKCD,Q;QAAA,2B;Q  
AAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAnCiD,WAmCvC,CAAY,OAAZ,C;UvBrpBP,U;UADP,YuBupBe,  
WvBvpBH,WuBupBwB,GvBvpBxB,C;UACL,IAAI,aAAJ,C;YACH,auBqpBuC,gB;YAA5B,WvBppBX,auBopBg  
C,GvBppBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuBipBA,iB;UACA,IAAK,WArCyD,cAqCrD,CAAe,OAAf,C  
AAJ,C;;QArCT,OAUco,W;O;KAnDX,C;yFAeA,yB;MAAA,+D;MAAA,sD;QAWoB,Q;QAAA,2B;QAAhB,OAA  
gB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UvBjoBP,U;UADP,YuBmoBe,WvBnoBH,WuBmoBwB  
,GvBnoBxB,C;UACL,IAAI,aAAJ,C;YACH,auBioBuC,gB;YAA5B,WvBhoBX,auBgoBgC,GvBhoBhC,EAAS,MA  
AT,C;YACA,e;;YAEA,c;;UuB6nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAhBX,C;yFAmBA,y  
B;MAAA,+D;MAAA,sE;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OA  
AZ,C;UvBrpBP,U;UADP,YuBupBe,WvBvpBH,WuBupBwB,GvBvpBxB,C;UACL,IAAI,aAAJ,C;YACH,auBqpBu  
C,gB;YAA5B,WvBppBX,auBopBgC,GvBppBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuBipBA,iB;UACA,IAAK  
,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAjBX,C;0FAoBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QA  
UW,sC;QAAA,8C;O;MAVX,oDAWQ,Y;QAA6C,OAAA,oBAAgB,W;O;MAXrE,iDAYQ,mB;QAAoC,gCAAY,O  
AAZ,C;O;MAZ5C,gF;MAAA,yC;QAUI,2D;O;KAVJ,C;IAGBA,sC;MASI,OAAO,yBAAqB,SAArB,EAA2B,SAA3  
B,C;K;IAGX,4C;MASI,OAAO,gCAA4B,SAA5B,EAakC,SAaIC,C;K;IAGX,mD;MASI,OAAoD,gBAA7C,gCAA  
4B,SAA5B,EAakC,SAaIC,CAA6C,C;K;4GAGxD,yB;MAuNA,wE;MAvNA,oD;QAGoiB,gB;QADb,YAAY,C;Q  
ACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAvNsB,U;UAAA,wBAuNT,oBAAmB,cAAAnB,EAAMb,sBAAnB,UA  
vNS,EAuNoB,IAvNpB,W;YAA6C,6B;;;QACHF,OAAO,W;O;KAVX,C;8FAaA,yB;MAAA,wE;MAAA,oD;QAUI  
B,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAAI,UAAU,oBAA  
mB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAC,IAAvC,CAAJ,C;;QACHB,OAAO,W;O;KAZX,C;IAeA,4C;MASI,O  
AA6C,gBAAcT,yBAAqB,SAArB,EAA2B,SAA3B,CAAsC,C;K;8FAGjD,yB;MAAA,oD;QA4KoB,Q;QAAA,2B;Q  
AAhB,OAAgB,cAAhB,C;UAAgB,yB;UArKK,U;UAAA,wBAqKQ,OArKR,W;YAAAsC,6B;;;QAC3D,OAAO,W;O;  
KARX,C;iFAWA,6C;MAOiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAA  
V,CAAJ,C;;MACHB,OAAO,W;K;IAGX,gC;MAOI,OAAO,qBAAiB,SAAjB,C;K;IAGB,6B;MAAE,S;K;IAX7B,+  
B;MAWI,OAAy,aAAL,SAAK,EAAW,eAAX,C;K;IAGhB,2C;MAYI,OAAO,qBAAiB,SAAjB,EAAuB,QAAvB,C;  
K;IAGX,mC;MASiB,Q;MADb,UAAU,sB;MACG,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,GAAl,WAAI,IAA  
J,C;;MACvB,OAAO,G;K;6EAGX,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IA  
AI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAQI,OAAO,oBAAW,U;K  
;6EAGtB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,  
C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;IAGX,6B;MAOoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,c  
AAhB,C;QAAgB,yB;QAAM,oBAAmB,qBAAnB,EAAMb,KAAAnB,E;;MACtB,OAAO,K;K;iFAGX,yB;MAAA,w  
E;MAAA,uC;QAOoB,Q;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAA  
U,OAAV,CAAJ,C;YAAwB,oBAAmB,qBAAnB,EAAMb,KAAAnB,E;;QAC9C,OAAO,K;O;KARX,C;8EAWA,yC;  
MAYoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAA  
V,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;4FAGX,yB;MAAA,wE;MAAA,gD;QAcOB,UAAiD,M;QAFjE,YAA  
Y,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAAn  
B,EAAMb,sBAAnB,UAAV,EAAC,WAAvC,EAAoD,OAApD,C;;QACpC,OAAO,W;O;KAFx,C;qFAkBA,6B;M  
AMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kGAG1B,yB;MAAA,w  
E;MAAA,oC;QASiB,UAAgC,M;QAD7C,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,o  
BAAmB,cAAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAAP,C,C;;O;KATvB,C;IAYA,2B;MAII,OAAO,uB;K;IAGX,2  
B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;iFAGX,+B;MAGW,sB;;QAYP,eAAe,oB;QACf,IAAI,CAAC,  
QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAp,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UA  
AyB,qBAAO,O;UAAp,uB;;QACzB,eAhBmB,QAGBJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAnBe,  
QAmBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAA  
T,QAAS,W;QACIB,qBAAO,O;;MAzBP,yB;K;6FAGJ,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q  
AAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHc,eAAe,  
SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;

UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACIB,OAAO,O;K;iFAGX,yB;MAAA,sE;MZpwC  
A,iB;MYowCA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,  
QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZhxCG,MAA  
O,KYgxCO,QZhxCP,EYgxCiB,CZhxCjB,C;;QYkxCd,OAAO,Q;O;KAtBX,C;iFAyBA,yB;MAAA,sE;MZxyCA,iB  
;MYwyCA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QA  
AS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZpzCG,MAAO,K  
YozCO,QZpzCP,EYozCiB,CZpzCjB,C;;QYszCd,OAAO,Q;O;KAtBX,C;iFAyBA,yB;MAAA,sE;MAAA,sC;QAaI,  
eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,  
OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,W  
AAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ/0CA,iB;MY+0CA,sC;QAaI,eAAe,oB;QACf,IAAI,CA  
AC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UA  
CI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZz1CG,MAAO,KYy1CO,QZz1CP,EYy1CiB,CZz1CjB,C;;QY21Cd,  
OAAO,Q;O;KApBX,C;6FAuBA,yB;MZj3CA,iB;MYi3CA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C  
;UAAyB,OAAO,I;QACChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,Q  
AAS,OAAIB,C;UACR,WZ33CG,MAAO,KY23CO,QZ33CP,EY23CiB,CZ33CjB,C;;QY63Cd,OAAO,Q;O;KApB  
X,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,Q  
AAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CA  
AX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;MAAA,kD;QAaI,eAAe,oB;QACf,IA  
AI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAA  
hB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,  
CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;qGAyBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,Q  
AAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,Q  
AAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;UA  
CI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OA  
AO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ18CG,MA  
AO,KY08CE,GZ18CF,EY08CO,CZ18CP,C;;MY48Cd,OAAO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,  
QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QA  
AS,O;QACjB,MZx+CG,MAAO,KYw+CE,GZx+CF,EYw+CO,CZx+CP,C;;MY0+Cd,OAAO,G;K;IAGX,iC;MAOI  
eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QA  
S,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;  
K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q  
AAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IA  
I,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,  
2B;MAII,OAAO,uB;K;IAGX,2B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;iFAGX,+B;MAGW,sB;;QAY  
P,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IA  
AI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eAhBmB,QAgBJ,CAAS,OAAT,C;;UAEX,Q  
AAQ,QAAS,O;UACjB,QAnBe,QAmBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;  
YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;;MAzBP,yB;K;6FAGJ,+B;MASI,eAAe,oB;MAC  
f,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAAd,C;  
QAAyB,OAAO,O;MACHC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QAC  
R,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACIB,OAAO,O  
;K;iFAGX,yB;MAAA,sE;MZj3CA,iB;MYi3CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,  
MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,O  
AAIB,C;UACR,WZ73CG,MAAO,KY63CO,QZ73CP,EY63CiB,CZ73CjB,C;;QY+3Cd,OAAO,Q;O;KAtBX,C;iFA  
yBA,yB;MAAA,sE;MZr5CA,iB;MYq5CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAA  
M,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB  
,C;UACR,WZj6CG,MAAO,KYi6CO,QZj6CP,EYi6CiB,CZj6CjB,C;;QYm6Cd,OAAO,Q;O;KAtBX,C;iFAyBA,yB;  
MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SA

AS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ57CA,iB;MY47CA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZt8CG,MAAO,KYs8CO,QZt8CP,EYs8CiB,CZt8CjB,C;;QYw8Cd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ99CA,iB;MY89CA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZx+CG,MAAO,KYw+CO,QZx+CP,EYw+CiB,CZx+CjB,C;;QY0+Cd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;qGAYBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZvjDG,MAAO,KYujDE,GZvjDF,EYujDO,CZvjDP,C;;MYyjDd,OAAO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZrIDG,MAAO,KYqlDE,GZrIDF,EYqlDO,CZrIDP,C;;MYuIDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KArBX,C;gGAWBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAAnB,EAAmB,oBAAAnB,QAaV,EAAuB,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KAtBX,C;4GAYBA,yB;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAAnB,EAAmB,oBAAAnB,QAaV,EAAuB,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KAtBX,C;8FAyBA,gC;MAGBI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;IAoBS,2I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;MAAA,gD;MAAA,kD;MAAA,wB;MAAA,+B;MAAA,kC;K;;;sDAAA,Y;;;cACZ,gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;uCACkB,0B;cACF,wD;cAAhB,gB;;;cAAA,KAAGB,yBAAhB,C;gBAAA,gB;;;cAAGB,oC;cACZ,yBAAc,6BAAU,sBAAV,EAAuB,OAAvB,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;K;IAPgB,wF;MAAA,yD;uBAAA,+H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAjBpB,sD;MAiBIOAAO,SAAS,iDAAT,C;K;IA4BS,yJ;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;MAAA,8D;MAAA,kD;MAAA,wB;MAAA,yB;MAAA,+B;MAAA,kC;K;;;6DAAA,Y;;;kBAKMc,I;cAJ/C,gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;iCACY,C;uCACM,0B;cACF,+D;cAAhB,gB;;;cAAA,KAAGB,yBAAhB,C;gBAAA,



gB;;;cAAgB,oC;cACZ,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAAnB,QAaV,EAaUc,sBAaVc,EAaOd  
,OAApD,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAIJ,W;;;;;;;  
K;IARgB,sG;MAAA,yD;uBAAA,6I;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAIBpB,6D;MAkBI,OAAO,SAAS,wD  
AAT,C;K;IA2BS,4H;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,oD;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,k  
C;K;;;wDAAA,Y;;;;oCACG,wC;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gCAAA,iCAA  
M,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;cAGI,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gB  
AAA,gB;;;cACI,yBAAc,6BAAU,sBAaV,EAaUc,mBAAS,OAAhC,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAA  
AA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAHJ,gB;;;cAQJ,W;;;;;;;K;IAVgB,yE;MAAA,yD;uBAAA,gH;YAA  
A,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,+C;MAgBI,OAAO,SAAS,0CAAT,C;K;IA6BS,0I;MAAA,wC;MAAA,6B  
;MAAA,yB;MAAA,kE;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,yB;MAAA,kC;K;;;+DAAA,Y;;;;cAOuC,Q;oC  
ANpC,+C;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2  
C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;cAGgB,C;cACZ,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;  
cACI,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAAnB,QAaV,EAaUc,sBAaVc,EAaOd,mBAAS,OAA7D  
,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAJJ,gB;;;cASJ,W;;;;;;;  
;K;IAXgB,uF;MAAA,yD;uBAAA,8H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,sD;MAgBI,OAAO,SAAS,iD  
AAT,C;K;IAcX,+C;MAkBI,OAAO,yBAAY,OAAZ,EAaQb,SAArB,C;K;IAGX,sD;MAmBI,OAAO,gCAAmB,OA  
AnB,EAa4B,SAa5B,C;K;gFAGX,+B;MASoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAA  
gB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;MAEJ,OAAO,G;K;4FAGX,+B;MASoB,Q;MADhB,UAAkB,G;MAC  
F,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;iFAGX,+B;MA  
YoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;M  
AEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q  
ACZ,YAAO,SAAS,OAAT,CAAP,I;MAEJ,OAAO,G;K;iFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;Q  
ADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO  
,G;O;KafX,C;iFakBA,yB;M3B15DA,6B;M2B05DA,sC;QAaoB,Q;QADhB,U3B55DmC,c2B45DnB,C3B55DmB,  
C;Q2B65DnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,M3BhuEiD,c2BguEjD,G3BhuE2D,KAAK,G2Bg  
uEzD,SAAS,OAAT,C3BhuEoE,KAAX,IAAf,C;;Q2BkuErD,OAAO,G;O;KAhBX,C;iFamBA,yB;MX16DA,+B;M  
W06DA,sC;QAaoB,Q;QADhB,UX36DqC,eAAW,oBW26D/B,CX36D+B,CAAX,C;QW46DrB,2B;QAAhB,OAAg  
B,cAAhB,C;UAAgB,yB;UACZ,MXhVEmD,eWgvEnD,GXhV8D,KAAK,KWgvE5D,SAAS,OAAT,CXhVeuE,KA  
AX,cAAhB,C;;QWkvEvD,OAAO,G;O;KAhBX,C;IAyBe,oD;MAAA,qB;QAAE,e;UAAM,MAAM,gCAAYB,2BA  
AwB,mBAAXB,MAAZB,C;;QAAZ,S;O;K;IANjB,qC;MAMI,OAAO,kBAAL,gCAAJ,C;K;IAGX,oC;MAaI,OAAO,  
sBAAS,IAAT,EAaE,IAAf,EAAsC,IAAtC,C;K;IAGX,+C;MAkBI,OAAO,sBAAS,IAAT,EAaE,IAAf,EAAsC,IAAt  
C,EAawD,SAAXD,C;K;IASA,0D;MAAA,4B;MAAA,sC;K;IAG0B,+E;MAAA,qB;QAAE,IAAI,CAAC,iBAAD,IA  
AY,WAAM,eAAN,cAAhB,C;UAAiC,oBAAU,I;UAA3C,OAAiD,K;;UAAjD,OAA8D,I;O;K;6CAF7F,Y;MACI,k  
BAAc,KAAc,C;MACA,OAAkB,SAAX,eAAW,EAaO,kEAAP,CAA8E,W;K;;IAT5G,qC;MAMI,kD;K;IASBo,6D;  
MAAA,wC;MAAA,4B;K;IAG6B,8D;MAAA,qB;QAAE,OAAM,aAN,mB;O;K;+CAFIC,Y;MACI,YAAqB,8BA  
AT,qBAAS,C;MACrB,OAAkB,YAAX,eAAW,EAaU,4CAAV,CAA0B,W;K;;IAjBxD,sC;MAaI,IAAI,Q9B80KG,  
YAAQ,C8B90Kf,C;QAAwB,OAAO,S;MAC/B,qD;K;IAqBO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;  
QAAE,OAAM,aAN,mB;O;K;+CALtC,Y;MACI,YAAqB,4BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QA  
CI,OAAO,eAAW,W;;QAEIB,OAAkB,YAAX,eAAW,EAaU,4CAAV,CAA0B,W;K;;IANB5D,sC;MAaI,qD;K;IAw  
BO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAAM,aAN,mB;O;K;+CALtC,Y;MACI,YAAq  
B,8BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,eAAW,W;;QAEIB,OAAkB,YAAX,eAAW,E  
AAU,4CAAV,CAA0B,W;K;;IANB5D,sC;MAaI,qD;K;8FAWJ,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,iBAAM,O  
AAN,C;O;KAPX,C;wFAUA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAYoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,g  
B;QACG,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAL,  
OAAJ,C;;YAEN,MAAO,WAAL,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KANBX,C;IASBa,oC;  
MAMI,OAA6C,UAAtC,YAAW,SAAX,EAaiB,YAAW,OAAx,EAajB,EAAsC,C;K;IAGjD,qC;MASI,OAAy,OA  
AL,SAAK,EAac,OAAT,QAAS,CAAd,C;K;IAGhB,qC;MASI,OAA+C,UAAx,C,YAAW,SAAX,EAa0B,aAAT,QA  
AS,CAA1B,EAawC,C;K;IAGnD,sC;MASI,OAAkC,UAA3B,YAAW,SAAX,EAaiB,QAAjB,EAa2B,C;K;4FAGt

C,yB;MAAA,0C;MAAA,qC;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;IAUA,2D;MAgB+C,oB;QAAA,OAAY,  
C;MAAG,8B;QAAA,iBAA0B,K;MACpF,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA2D,KAA3  
D,C;K;IAGX,sE;MAkBkD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,OAAwE,OAAjE,8BAAiB  
,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA2D,IAA3D,CAAiE,EAAI,SAAJ,C;K;IAYpC,4B;MAAY,cAAM,EA  
AN,C;K;IATpD,kC;MASI,OAAO,oBAAgB,SAAhB,EAAsB,KAAtB,EAA6B,UAA7B,C;K;IAGX,6C;MAUI,OAA  
O,oBAAgB,SAAhB,EAA6B,SAAJ7B,C;K;IacY,kC;MAAU,aAAK,CAAL,C;K;IAXjC,kC;MAWI,  
OAAO,yBAAY,kBAAZ,C;K;IAeiB,wH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,gD;MAAA,kD;MAAA,4B;M  
AAA,2B;MAAA,wB;MAAA,kC;K;;;sDAAA,Y;;;oCACL,sC;cACf,IAAI,CAAC,mBAAS,UAAAd,C;gBAAYB,M;;  
gBAAZB,gB;;;;;mCACc,mBAAS,O;cACvB,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;gCACe,mBAA  
S,O;cACpB,gB;8BAAA,iCAAM,6BAAU,kBAAV,EAAmB,eAAnB,CAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;c  
ACA,qBAAU,e;cAHd,gB;;;cAKJ,W;;;K;IATwB,uE;MAAA,yD;uBAAA,4G;YAAA,S;iBAAA,Q;;iBAAA,  
uB;O;K;IAZ5B,6C;MAYI,OAAO,SAAS,0CAAT,C;K;IAYX,8F;MAU6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA  
,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QA  
AA,YAAoC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAh  
B,C;QAAGB,yB;QACZ,IAAI,CAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IA  
Aa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CA  
AT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,  
M;K;IAGX,4F;MAUwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,  
qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,E  
AAwB,SAAXB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAAJ3D,EAA6B,SAAtE,CAAiF,W;  
K;IAOXE,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAJ3B,kC;MAII,oCAAGB,8BAAhB,C;K;2FAGJ,qB;MAK  
I,OAAO,S;K;IAGX,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAGB,cAAhB,C;  
QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAAAS,CAAb,GAAg  
B,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;  
MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,  
OAAW,UAAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;  
MACIB,YAAiB,C;MACD,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAn  
B,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;M  
ASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAA  
O,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAAAS,CAAb,GAAgB,wCAAO,IAAvB,GAA  
gC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAGB,cAAh  
B,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAAAS,CAAb,G  
AAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MAC  
D,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;M  
AEJ,OAAW,UAAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAQoB,Q;MADhB,UAAe,  
C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q  
;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAG  
X,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,OAAP,I;;  
MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,  
cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAGB,cAAh  
B,C;QAAGB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAkB,G;MACF,2B;MAA  
hB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IC71FX,qC;MAMI,aAAa,qBAAiB,YAA  
Y,cAAZ,CAAjB,C;MACb,kBAAC,KAAd,C;MX4zBgB,Q;MAAA,OW3zBT,SX2zBS,W;MAAhB,OAAGB,cAAhB  
,C;QAAGB,2B;QAAU,oB;QW3zBK,IAAI,CAAC,SAAD,IAAY,OX2zBX,SW3zBW,UAAhB,C;UAAiC,YAAU,I;  
UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QX2zBvE,qB;UW3zBD,MX2zBqC,WAAI,SAAJ,C;;MW3zB1D,OAA  
qB,M;K;IAGzB,sC;MAUI,aAAa,qBAAiB,SAAJB,C;MACN,YAAP,MAAO,EAAU,QAAV,C;MACP,OAAO,M;K;  
IAGX,sC;MAUI,YAAqB,gCAAT,QAAS,EAAgC,SAAhC,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAY,QAAL  
,SAAK,C;MACHB,IAAI,yBAAJ,C;QACgB,kBAAY,sB;QXixBZ,Q;QAAA,OWjxBL,SXixBK,W;QAAhB,OAAGB

,cAAhB,C;UAAgB,yB;UAAM,IAAI,CWjxBwB,qBXixBb,OWjxBa,CXixB5B,C;YAAyB,WAAY,WAAI,OAAJ,C;  
;QWjxBvD,OXkxBG,W;;MWjxBP,aAAa,qBAAiB,SAAjB,C;MACb,MAAO,mBAAU,KAAV,C;MACP,OAAO,M;  
K;IAGX,uC;MAUI,aAAa,qBAAiB,SAAjB,C;MACN,YAAP,MAAO,EAAU,QAAV,C;MACP,OAAO,M;K;gGAG  
X,yB;MAAA,8C;MAAA,qC;QAOI,OAAO,iBAAM,OAAN,C;O;KAPX,C;IAUA,qC;MAMI,aAAa,qBAAiB,YAA  
Y,iBAAO,CAAP,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAA  
O,M;K;IAGX,sC;MAOI,aAAa,qBAAiB,YAA,Y,SAAK,KAAL,GAAY,QAAS,OAAR,B,IAAZ,CAAjB,C;MACb,MA  
AO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;MAMuD,UAAT,M;  
MAAIC,aAAa,qBAAiB,YAA,Y,WAAS,4BAAT,QAAS,CAAT,YAA4C,cAAL,WAAvC,4BAA2D,SAAK,KAAL,  
GAAY,CAAZ,IAAvE,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,  
OAAO,M;K;IAGX,sC;MAOI,aAAa,qBAAiB,YAA,Y,SAAK,KAAL,GAAY,CAAZ,IAAZ,CAAjB,C;MACb,MAA  
O,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;8FAGX,yB;MAAA,4C;MAAA,q  
C;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;InBnIA,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3  
C,gF;ICGA,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;gGmBYA,yB;MAAA,uD;MAAA,gC;M  
AAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KA AJ,CAAtC,GAAsD,uBAAa,KA  
Ab,E;O;KAPjE,C;gGAUA,yB;MAAA,+C;MAAA,mC;QAOI,OAA,Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;  
0EAUA,yB;MA4EA,6C;MAAA,oC;MAAA,gC;MA5EA,uC;QAOW,sB;;UAyES,Q;UAAA,0B;UAAhB,OAAGB,c  
AAhB,C;YAAgB,oC;YAAM,IAzEH,SAyEO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qB  
AAO,I;;;QA1EP,yB;O;KAPJ,C;kFAUA,yB;MAwJA,mD;MAAA,+C;MAAA,oC;MAxJA,uC;QAOW,qB;;;UAuJO,  
Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAC,cAAd,C;YAAc,uB;YACV,cAAc,qBAAK,KAAL,C;YAC  
d,IAzJc,SAyJV,CAAU,oBAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA3JP,wB;O;KAPJ,  
C;IAUA,6B;MAKI,ICkOgD,qBAAU,CDIO1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAK,CA  
AL,C;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,0B;QAAhB,OA  
AgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,  
6DAAvB,C;O;KANV,C;6FASA,yB;MAAA,iE;MAYA,6C;MAAA,oC;MAAA,gC;MAZA,uC;QASW,Q;QAAA,+B  
;;UAYS,U;UAAA,4B;UAAhB,OAAGB,gBAAhB,C;YAAgB,sC;YACZ,aAbwB,SAaX,CAAU,oBAAV,C;YACb,IA  
AI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIBA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,sEA  
AvB,C;;QAAD,OAAO,I;O;KATX,C;yGAYA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QASoB,Q;QAA  
A,0B;QAAD,OAAGB,cAAhB,C;UAAgB,oC;UACZ,aAAa,UAAU,oBAAV,C;UACb,IAAI,cAAJ,C;YACI,OAAO,  
M;;;QAGf,OAAO,I;O;KafX,C;IAkBA,mC;MAII,OckLgD,qBAAU,CDILnD,GAAe,IAAf,GAAYB,qBAAK,CAAL  
,C;K;wFAGpC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAIOB,Q;QAAA,0B;QAAD,OAAGB,cAAhB,C  
;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;mFAQA,yB  
;MAAA,uD;MAAA,gC;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KA AJ,CA  
AtC,GAAsD,uBAAa,KAAb,E;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAsC  
,qBAAI,KA AJ,CAAtC,GAAsD,I;K;0FAGjE,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAikB,gC;QAAA,6B;QAAA,  
mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,  
OAAO,E;O;KATX,C;wFAYA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR  
,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,O  
AAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,4B;MAQI,ICsHgD,qBAAU,CDtH1D,C;QACI,MAAM,2BAAuB,yB  
AAvB,C;MACV,OAAO,qBAAK,2BAAL,C;K;0EAGX,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,  
uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAA  
c,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,6DAAv  
B,C;O;KAZV,C;IAeA,kC;MAMI,OC4FgD,qBAAU,CD5FnD,GAAe,IAAf,GAAYB,qBAAK,mBAAS,CAAT,IAAL  
,C;K;sFAGpC,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,  
CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,C  
AAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;8EAaA,yB;MAAA,mC;MAAA,yC;MAAA,4B;QAQI,O  
AAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,IC0DgD,qBAAU,CD1D1D,C;QACI,MAAM,2BAAuB,yBA  
AvB,C;MACV,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;0FAGX,yB;MAAA,mC;MAAA,qD;MAAA,4  
B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,ICqCgD,qBAAU,CDrC1D,C;QACI,OAAO,I;MA

CX,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aA  
CH,C;UAAK,MAAM,2BAAuB,yBAAvB,C;aACX,C;UAAK,4BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAY  
B,0CAAzB,C;;MAHIB,W;K;8EAOJ,yB;MAAA,6C;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MA  
AA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;U  
ACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAJ,C;cAAW,MAAM,8BAAyB,wDAAzB,C;YACjB,SAAS,O;  
YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,6DAAvB,C;QAEIB,OAAO,4E;O;Kaf  
X,C;IAkBA,oC;MAII,OAAW,qBAAU,CAAd,GAAiB,qBAAK,CAAL,CAAjB,GAA8B,I;K;0FAGzC,yB;MAAA,6  
C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,c  
AAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAJ,C;cAAW,OAAO,I;YACIB,SAAS,  
O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,OAAO,I;QACnB,OAAO,M;O;KadX,C;IAiBA,+B;MI  
BzRI,IAAI,EkBiSI,KAJK,CIBjST,CAAJ,C;QACI,ckBgSc,wD;QIB/Rd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkB  
gSV,OAAO,8BAAC,eAAF,CAAE,EAAa,gBAAb,CAAd,EAAoC,gBAAP,C,C;K;IAGX,+B;MIBrSI,IAAI,EkB6SI,K  
AAK,CIB7ST,CAAJ,C;QACI,ckB4Sc,wD;QIB3Sd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkB4SV,OLhH6E,oBKg  
HID,eAAF,CAAE,EAAa,gBAAb,CLhH0D,C;K;IKmHjF,kC;MIBjTI,IAAI,EkByTI,KAJK,CIBzTT,CAAJ,C;QAC  
I,ckBwTc,wD;QIBvTd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBwTV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT  
,IAAY,EAAc,CAAd,CAAIb,C;K;IAGX,mC;MIB7TI,IAAI,EkBqUI,KAJK,CIBrUT,CAAJ,C;QACI,ckBoUc,wD;  
QIBnUd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBoUV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,  
CAAd,CAAIb,C;K;2FAGX,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UA  
CI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAy,CAAZ,EAAe,QAAQ,CAAR,IAAf,  
C;QACf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CA  
A/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OL5JoF,oBK4JnE,CL5JmE,EK4JhE,QA  
AQ,CAAR,IL5JgE,C;;QK6J5F,OAAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UA  
AL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI  
,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAy,KAAZ,EAAmB,gBAAnB,C;QACf,O  
AAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAL,MAAK,EAAL,MAAK,EAAL  
,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KA  
AL,EAAV,CAAL,C;YACI,OLvLqE,oBKuLpD,KLvLoD,C;;QKwL7E,OAAO,E;O;KATX,C;8EAYA,yB;MAAA,y  
D;MAkFA,oC;MAIFA,uC;QAMW,kBAAS,oB;QAKFM,Q;QAAA,uB;QAAtB,iBAAC,CAAd,wB;UACI,cAAc,qBA  
AI,KAJ,C;UACd,IApF6B,SAoFzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QApFxC,OAsF  
O,W;O;KA5FX,C;8EASA,yB;MAAA,yD;MAyEA,oC;MAzEA,uC;QAMW,kBAAS,oB;QAYEM,Q;QAAA,uB;QA  
AtB,iBAAC,CAAd,wB;UACI,cAAc,qBAAI,KAJ,C;UACd,IA3E6B,SA2EzB,CAAU,oBAAV,CAAJ,C;YAAwB,  
WAAy,gBAAO,OAAP,C;;QA3ExC,OA6EO,WA7EqC,W;O;KANhD,C;4FASA,yB;MAAA,yD;MASBA,gC;MA+  
sBA,6C;MAAA,oC;MARuBA,uC;QAQW,kBAAGB,oB;QAouBV,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cA  
Ab,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UA7sB/B,IAvBoC,SAuBhC,CAAU,OA  
AV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAvB/C,OAYBO,W;O;KAjCX,C;4FAWA,yB;MAAA,y  
D;MAWA,gC;MA+sBA,6C;MAAA,oC;MAItBA,uC;QAQW,kBAAGB,oB;QAYtBV,gB;QADb,YAAY,C;QACC,0  
B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UA7sB/B,IAZoC,SA  
YhC,CAAU,OA AV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAZ/C,OAcO,WAd4C,W;O;KARvD,C  
;gGAWA,yB;MAAA,gC;MA+sBA,6C;MAAA,oC;MA/sBA,oD;QAstBiB,gB;QADb,YAAY,C;QACC,0B;QAAb,O  
AAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UA7sB/B,IAAI,UAAU,OA AV,  
EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAE/C,OAAO,W;O;KAXX,C;oFAcA,yB;MAAA,yD;MAK  
BA,6C;MAAA,oC;MAAA,gC;MAIBA,uC;QAMW,kBAAY,oB;QAKBH,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;  
UAAgB,oC;UAAM,IAAI,CAIBU,SAkBT,CAAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAIB3D,  
OAmBO,W;O;KAZBX,C;oFASA,yB;MAAA,yD;MASA,6C;MAAA,oC;MAAA,gC;MATA,uC;QAMW,kBAAY,o  
B;QASH,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CATU,SAST,CAAU,oBAAV,CAAL  
,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAT3D,OA UO,WAVwC,W;O;KANnD,C;wFASA,yB;MAAA,6C;MAAA,  
oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,U  
AAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAC3D,OAAO,W;O;KAPX,C;kFAUA,yB;MAAA,o



K;IAGX,iC;MAII,OAAO,2BAAa,eAAc,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB,CAAd,CAAb,C;K;IAGX,8  
B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,iCAAK,CAAL  
,EAAP,C;UAAL,K;;UACa,wBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,qC;MAII,OAAO,2BAAa,iBAAgB,gB  
AAhB,CAAb,C;K;IAGX,6B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,C;U  
AAK,aAAM,iCAAK,CAAL,EAAN,C;UAAL,K;;UACQ,kCAAa,qBAAoB,YAAmB,eAAP,gBAAO,EAAa,GAAb,  
CAAnB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;gFAOJ,yB;MAAA,+D;MA0CA,6C;MAAA,oC;MAAA,gD;MA  
AA,gC;MA1CA,uC;QAMW,kBAAU,gB;QA wCD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,W  
AzC6B,SAyCIB,CAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1ChB,OA4CO,W;O;KAIDX,C;8FA  
SA,yB;MAAA,+D;MAeA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAfA,uC;QAYW,kBAAiB,gB;QAcR,gB;QADh  
B,YAAY,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,Wafoc,SAezB,EAAU,cAAV,EAAU,sBAA  
V,WAAmB,oBAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhBhB,OakBO,W;O;KA9BX,C;kGAeA,yB;M  
AAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,0B;QAAhB,O  
AAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,  
WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;oFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,g  
C;MAAA,oD;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,UAAU,oBAAV,C;UA  
CC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;gFAWA,yB;MAAA,wE;MAyBA,6C;MAAA,o  
C;MAAA,+D;MAAA,gC;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAA  
gB,oC;UACZ,UA1BoD,WA0B1C,CAAY,oBAAZ,C;UzBrjBP,U;UADP,YyBujBe,WzBvjBH,WyBujBwB,GzBvjB  
xB,C;UACL,IAAI,aAAJ,C;YACH,ayBqjBuC,gB;YAA5B,WzBpjBX,ayBojBgC,GzBpjBhC,EAAS,MAAT,C;YAC  
A,e;;YAEA,c;;UyBijBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5BT,OA8BO,W;O;KA vCX,C;gFAYA,yB;MAAA,  
wE;MA8BA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,0B;QAAh  
B,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UA/BiD,WA+BvC,CAAY,oBAAZ,C;UzBvkBP,U;UADP,YyBykBe,WzB  
zkBH,WyBykBwB,GzBzkBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBukBuC,gB;YAA5B,WzBtkBX,ayBskBgC,GzBt  
kBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBmkBA,iB;UACA,IAAK,WAjCyD,cAiCrD,CAAE,oBAAf,CAAJ,C;  
;QAJCT,OAmCO,W;O;KA7CX,C;oFAaA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q  
;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;UzBrjBP,U;UADP,YyBujBe  
,WzBvjBH,WyBujBwB,GzBvjBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBqjBuC,gB;YAA5B,WzBpjBX,ayBojBgC,G  
zBpjBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBijBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;K  
AdX,C;oFAiBA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAAhB,OA  
AgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;UzBvkBP,U;UADP,YyBykBe,WzBzkBH,WyBykBw  
B,GzBzkBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBukBuC,gB;YAA5B,WzBtkBX,ayBskBgC,GzBtkBhC,EAAS,MA  
AT,C;YACA,e;;YAEA,c;;UyBmkBA,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,  
C;qFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDAS  
Q,Y;QAAGD,OAAgB,SAAhB,oBAAgB,C;O;MATxE,iDAUQ,mB;QAAuC,gCAAY,oBAAZ,C;O;MAV/C,gF;MA  
AA,yC;QAQI,2D;O;KARJ,C;wEAeA,yB;MAAA,gE;MAyEA,6C;MAAA,oC;MAAA,gC;MAzEA,uC;QAOW,kBA  
AM,eAAa,gBAAb,C;QAUeA,Q;QAAA,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAXEmB,SAwEf,CA  
AU,iBAAV,CAAJ,C;;QAxehB,OAYEO,W;O;KAhFX,C;sFAUA,yB;MAAA,gE;MA+BA,6C;MAAA,oC;MAAA,g  
C;MA/BA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAGCP,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UA  
Aa,iC;UACT,WAAY,WAjC0B,SAiCtB,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAJChB,OAK  
CO,W;O;KAZCX,C;mGAUA,yB;MAAA,+D;MAUA,gC;MAoLA,6C;MAAA,oC;MA9LA,uC;QAOW,kBAAoB,g  
B;QA8Ld,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UApLsB,U;UAAA,cAVQ,SAUR,EAo  
LT,cApLS,EAoLT,sBApLS,WAO LA,iBApLA,W;YAA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;uGAUA,yB;MAA  
A,gC;MAoLA,6C;MAAA,oC;MApLA,oD;QA2LiB,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,i  
C;UApLsB,U;UAAA,yBAoLT,cApLS,EAoLT,sBApLS,WAO LA,iBApLA,W;YAA6C,6B;;;QACHF,OAAO,W;O;K  
ARX,C;0FAWA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAQiB,UACiB,M;QAF9B,YAAY,C;QACC,0  
B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ  
,C;;QACHB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,+D;MAUA,gC;MA2IA,6C;MAAA,oC;MARJA,uC;QAOW,  
kBAAa,gB;QAKJJ,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA1IK,U;UAAA,cARe,SAQf,CA0IQ,oB

A1IR,W;YAAAsC,6B;;;QAR3D,OASO,W;O;KAhBX,C;yFAUA,yB;MAAA,gC;MA2IA,6C;MAAA,oC;MA3IA,oD;  
QA+IoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA1IK,U;UAAA,wBA0IQ,oBA1IR,W;YAAAsC,6B  
;;;QAC3D,OAAO,W;O;KANX,C;4EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAKiB,Q;QAAA,0B;  
QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAy,WAAl,UAAU,iBAAV,CAAJ,C;;QAChB,OAAO,W;O;KAPX,C;I  
Ae4B,4C;MAAA,mB;QAAE,iC;O;K;IAL9B,iC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;wEAGX,yB;MAAA,6C;MA  
AA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAA  
C,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAPX,C;IAUA,2B;MAMI,OAAO,ECrwByC,  
qBAAU,CDqwBnD,C;K;wEAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAh  
B,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;  
KAPX,C;4EUAU,qB;MAKI,OAAO,gB;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAKoB,Q;Q  
ADhB,YAAy,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAw  
B,qB;;QAC9C,OAAO,K;O;KANX,C;0EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADh  
B,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,UAAU,WAAV,EAAuB,oBAAvB,  
C;;QACpC,OAAO,W;O;KAXX,C;wFACa,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAYoB,UAA8B,M;  
QAF9C,YAAy,C;QACZ,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,WAAU,cA  
AV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,oBAAhC,C;;QACpC,OAAO,W;O;KAbX,C;mFAGBA,yB;MAAA,  
uD;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAAy,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;U  
ACI,cAAc,UAAU,kCAAI,YAAJ,EAAI,oBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAdX,C;iGaiB  
A,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAUI,YAAy,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;U  
ACI,cAAc,UAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhB  
X,C;gFAMBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;  
UAAgB,oC;UAAM,OAAO,oBAAP,C;;O;KAJ1B,C;8FAOA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QA  
OiB,UAAa,M;QAD1B,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP  
,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,2B;MAGI,OAAO,uB;K;4EAGX,yB;MAMA,uD;MAAA,oC;MANA,sC;  
QAGW,sB;;UAUP,ICz4BgD,qBAAU,CDy4B1D,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,qBAAK,CAAL,C;U  
ACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAJB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAdmB,QAcJ,CAAS,oB  
AAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,qBAAK,CAAL,C;YACR,QAjBe,QAiBP,CAAS,cAAT  
,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;;QAvBP,yB;O;  
KAHJ,C;wFAMA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICz4BgD,qBAAU,CDy4B1D,C;UAAe,OAAO,I;Q  
ACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAJB,C;UAAoB,OAAO,O;  
QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QA  
AQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O  
;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdzncA,iB;McynCA,sC;QAEiB,Q;QAFb,ICt6BgD,  
qBAAU,CDs6B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,C  
AAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdloCG,MAAO,KckoCO,QdloCP,EckoCiB,CdloCj  
B,C;;QcooCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;Md1pCA,iB;Mc0pCA,sC;Q  
AeiB,Q;QAFb,IC57BgD,qBAAU,CD47B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdnqCG,MAAO,KcmqC  
O,QdnqCP,EcmqCiB,CdnqCjB,C;;QcqqCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD  
;MAAA,sC;QAaiB,Q;QAFb,IC9B9BgD,qBAAU,CDg9B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CA  
AL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAA  
W,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;Md3r  
CA,iB;Mc2rCA,sC;QAaiB,Q;QAFb,ICt+BgD,qBAAU,CDs+B1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,  
CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdlsCG  
,MAAO,KcksCO,QdlsCP,EcksCiB,CdlsCjB,C;;QcosCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,u  
D;Md1tCA,iB;Mc0tCA,sC;QAaiB,Q;QAFb,IC1/BgD,qBAAU,CD0/B1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,i  
CAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,  
WdjuCG,MAAO,KciuCO,QdjuCP,EciuCiB,CdjuCjB,C;;QcmuCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;

MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,IC5gCgD,qBAAU,CD4gC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iC  
AAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IA  
AI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;ofAoBA,yB;MAAA,sE;MAAA,o  
C;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,ICliCgD,qBAAU,CDkiC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,i  
CAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,I  
AAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K  
AnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICtjCgD,qBAAU,CDsjC1D,C;UAAe,  
OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC  
AAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WA  
AW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICtkCgD,qBAAU,CDskC1D,C;QAAe,OAA  
O,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR  
,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAGI,OAAO,4BAAc,UAAc,C;K;IAG  
X,iD;MAOiB,Q;MAFb,IC1lCgD,qBAAU,CD0lC1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,  
kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAaa,cAAb,C  
AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAGI,OAAO,uB;K;4EAGX,yB;MA  
MA,uD;MAAA,oC;MANA,sC;QAGW,sB;;UAUP,ICtnCgD,qBAAU,CDsnC1D,C;YAAe,qBAAO,I;YAAP,uB;;UA  
Cf,cAAc,qBAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UA  
CpB,eAdmB,QAcJ,CAAS,oBAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,qBAAK,CAAL,C;YAC  
R,QAjBe,QAiBP,CAAS,cAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAG  
nB,qBAAO,O;;;QAvBP,yB;O;KAHJ,C;wFAMA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICtnCgD,qBAAU,C  
DsnC1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,  
CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,  
qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,  
WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdlpCA,iB;MckpCA  
,sC;QAEiB,Q;QAFb,ICnpCgD,qBAAU,CDmpC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EA  
AT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd3pCG,MAAO,  
Kc2pCO,Qd3pCP,Ec2pCiB,Cd3pCjB,C;;Qc6pCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MA  
AA,uD;MdnrCA,iB;McmrCA,sC;QAEiB,Q;QAFb,ICzqCgD,qBAAU,CDyqC1D,C;UAAe,MAAM,6B;QACrB,eAA  
e,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C  
;UACR,Wd5rCG,MAAO,Kc4rCO,Qd5rCP,Ec4rCiB,Cd5rCjB,C;;Qc8rCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MA  
AA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,IC7rCgD,qBAAU,CD6rC1D,C;UAAe,MAAM,6B;QAC  
rB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,  
EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;  
MAAA,oC;MAAA,uD;MdptCA,iB;McotCA,sC;QAaiB,Q;QAFb,ICntCgD,qBAAU,CDmtC1D,C;UAAe,OAAO,I;  
QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CA  
AL,EAAT,C;UACR,Wd3tCG,MAAO,Kc2tCO,Qd3tCP,Ec2tCiB,Cd3tCjB,C;;Qc6tCd,OAAO,Q;O;KAjBX,C;wFAo  
BA,yB;MAAA,oC;MAAA,uD;MdnvCA,iB;McmvCA,sC;QAaiB,Q;QAFb,ICvuCgD,qBAAU,CDuuC1D,C;UAAe,  
OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC  
AAK,CAAL,EAAT,C;UACR,Wd1vCG,MAAO,Kc0vCO,Qd1vCP,Ec0vCiB,Cd1vCjB,C;;Qc4vCd,OAAO,Q;O;KAj  
BX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,ICzvCgD,qBAAU,CDyvC1D,C;UAAe,O  
AAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCA  
AK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;of  
AoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,IC/wCgD,qBAAU,CD+wC1D,C;UAAe,  
MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS  
,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W  
AAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICny  
CgD,qBAAU,CDmyC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAA  
U,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,C



AAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICnzCgD ,qBAAU,CDmzC1D,C;QAAe,OAAO,I;MAcTb,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QA CI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;M AGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOiB,Q;MAFb,ICv0CgD,qBAAU,CDu0C1D,C;QAAe,OAAO,I;MA CtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI, UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4 B;MAMI,OCt1CgD,qBAAU,C;K;0EDy1C9D,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAA A,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;;QACrD, OAAO,I;O;KAPX,C;8EAUA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAKmC,Q;QAAA,0B;QAAhB,O AAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;QAARc,gB;O;KALJ,C;4FAQA,yB;MAAA,6B;MAAA,sC ;MA/fA,6C;MAAA,oC;MAAA,gC;MA+fA,2BAQiB,yB;QAvGbjB,6C;QAAA,oC;QAAA,gC;eAugBiB,0B;UAAA, 4B;YAAE,aAAe,c;YAhGbjB,gB;YADb,YAAY,C;YACC,0B;YAAb,OAAa,cAAb,C;cAAa,iC;cAAM,QAAO,cAAP ,EAAO,sBAAP,WAAgB,iBAAhB,C;;YAggBmB,W;W;S;OAAzB,C;MARjB,oC;QAxfiB,gB;QADb,YAAY,C;QA CC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QAggBnB,gB; O;KARJ,C;8EAWA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICn4CgD,q BAAU,CDm4C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB ,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KAnBX ,C;4FAsBA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICz5CgD,qBAAU, CDy5C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc, CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O; KAnBX,C;wGAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC/6CgD,qBAAU,CD +6C1D,C;UACI,OAAO,I;QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,o BAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;0FAsBA,yB; MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAIbqB,Q;QAHjB,ICt8CgD,qBAAU,CDs8C1D,C;UACI,OAAO,I; QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,i CAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KApBX,C;uFAuBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA, gC;MAAA,uC;QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCA A9B,C;QACrB,kBAakB,sBAAL,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,k CAAI,CAAJ,EAAI,sBAAL,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O;KAnBX,C;qGAsBA,yB;MAAA,uD ;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe, MAAM,mCAA8B,uCAA9B,C;QACrB,kBAakB,sBAAL,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C; UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;K ApBX,C;iHAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,Q AAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAL,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C; UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;K ApBX,C;mGAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QA CZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAL,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS ,CAAhB,C;UACI,cAAc,oBAAU,kCAAI,CAAJ,EAAI,sBAAL,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O; KApBX,C;wFAuBA,yB;MAAA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAgBoB,Q;QAHh B,ICvjDgD,qBAAU,CDujD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C ;QAAiC,8B;QAA9C,af5wDO,W;Qe6wDP,kBAakB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,c AAAC,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;sGAuB A,yB;MAAA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAAA,gD;QAIbKb,gC;QAHd,IC/kDgD,qBAAU,CD+kD1D ,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,afpyDO, W;Qe9yDP,kBAakB,O;QACJ,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAi B,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4FA wBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,Q;QAHtB,ICtmDgD,qBAAU,CDsm D1D,C;UAAe,OAAO,W;QACtB,sBAakB,qBAAK,CAAL,CAAlB,C;QACqC,kBAAXB,eAAgB,gBAAhB,C;QAA

gC,sBAAl,0BAAJ,C;QAA7C,af5zDO,W;Qe6zDe,uB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,oBAAU,0BAAV,EAAuB,iCAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KApBX,C;0GAuBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAIb0B,Q;QAHtB,IC9nDgD,qBAAU,CD8nD1D,C;UAAe,OAAO,W;QACtB,sBAaKB,qBAaK,CAAL,CAAlB,C;QACqC,kBAAXB,eAAgB,gBAAhB,C;QAAgC,sBAAl,0BAAJ,C;QAA7C,afp1DO,W;Qeq1De,uB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,oBAAU,KAAV,EAAiB,0BAAJB,EAA8B,iCAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0EAWBA,yB;MA9FA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MA8FA,gD;QAcW,sB;;UA5FS,Q;UAHhB,ICvjDgD,qBAAU,CDujD1D,C;YAAe,qBAAO,OA+FH,OA/FG,C;YAAP,uB;;UACuB,kBAAZB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBA8F3B,OA9F2B,C;UAA9C,af5wDO,W;Ue6wDP,kBA6FmB,O;UA5FH,0B;UAAhB,OAAgB,cAAhB,C;YAAgB,oC;YACZ,cA2FwB,SA3FV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAwFP,yB;O;KAdJ,C;wFAiBA,yB;MAxFA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAwFA,gD;QAEW,6B;;UAtFO,gC;UAHd,IC/kDgD,qBAAU,CD+kD1D,C;YAAe,4BAAO,OAyFI,OAzFJ,C;YAAP,8B;;UACuB,kBAAZB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBAwFpB,OAxFoB,C;UAA9C,afpyDO,W;UeqyDP,kBAuF0B,O;UAtFZ,6B;UAAA,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAQf+B,SArFjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAKFP,gC;O;KAFJ,C;4EAKBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;wFAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAVX,C;4EAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACJ,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,SASoB,gB;MATpB,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACgB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M7BppDA,6B;M6BopDA,sC;QAWoB,Q;QADhB,U7BppDmC,c6BopDnB,C7BppDmB,C;Q6BqpDnB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,M7Bx9DiD,c6Bw9DjD,G7Bx9D2D,KAAK,G6Bw9DzD,SAAS,oBAAT,C7Bx9DoE,KAAAX,IAAf,C;;Q6B09DrD,OA AO,G;O;KAdX,C;4EAIbA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MblqDA,+B;MakqDA,sC;QAWoB,Q;QADhB,UbjqDqC,eAAW,oBaiqD/B,CbjqD+B,CAAX,C;QakqDrB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,Mbt+DmD,eas+DnD,Gbt+D8D,KAAK,Kas+D5D,SAAS,oBAAT,Cbt+DuE,KAAAX,CAAhB,C;;Qaw+DvD,OAAO,G;O;KAdX,C;IAiBA,oC;MAWI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,+C;MAGBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,EAAwD,SAAXD,C;K;IAcSB,oC;MAAE,OAAA,EAAG,W;K;IAXtC,0C;MAWI,OAAO,6BAAgB,IAAhB,EAAcB,sBAAtB,C;K;IAGX,uD;MAGBI,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EA A8C,IAA9C,EAAgE,SAAhE,C;K;oFAGX,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,u C;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI ,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAAO,cAA K,KAAL,EAAy,MAAZ,C;O;KAjBX,C;oFAoBA,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;M AAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAC Z,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAA O,cAAK,KAAM,WAAAX,EAAuB,MAAO,WAA9B,C;O;KAjBX,C;IAqCgD,6B;MAAE,OAAA,EAAG,W;K;IAjBr D,2D;MAGB4C,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACjF,OAAO,sBAAS,IAAT,EAAe,IAAf,E AAqB,cAArB,EAAqC,eAArC,C;K;IAGX,sE;MAkBgD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAQ hE,Q;MAPrB,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,eAAe,SAAK,O;MACpB,qBAAqB,YAAW,IAAX,SAAS B,WAAW,IAAX,KAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;MACrB,aAAa,iBAAa,cAAb,C;MACb,Y AAY,C;MACZ,OAAgB,CAAT,qBAAiB,QAAXB,C;QACI,UAAU,QAAQ,IAAR,I;QACO,IAAI,MAAM,CAAN,IA AW,MAAM,QAAR,C;UAAiC,IAAI,cAAJ,C;YAAoB,e;;YAAc,K;;UAAa,U;QAAjG,qB;QACA,MAAO,WAAI,U AAU,8BAAy,KAAZ,EAAMB,UAAAnB,CAAV,CAAJ,C;QACP,gBAAS,IAAT,I;;MAEJ,OAAO,M;K;IAoB6C,qC; MAAE,OAAA,EAAG,W;K;IAjB7D,iE;MAGBoD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACzF,O

AAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA6C,uBAA7C,C;K;IAwByB,2F;MAAA,wB;QAC5B,U  
AAU,QAAQ,YAAR,I;QACV,iBAAqB,MAAM,CAAN,IAAW,MAAM,4BAArB,GAA6B,4BAA7B,GAAyC,G;QA  
D1D,OAEA,kBAAU,0CAA,Y,KAaZ,EAAM,B,UAAAnB,CAAV,C;O;K;IAxBR,gF;MAkBWd,sB;QAAA,SAAY,C;  
MAAG,8B;QAAA,iBAA0B,K;MAC7F,oBAAoB,IAApB,EAA0B,MAA1B,C;MACA,cAAc,KAAK,cAAJ,GAAoB,  
yBAApB,GAAiC,WAAQ,mBAAS,IAAT,GAAgB,CAAhB,IAAR,CAAIC,EAAkE,MAAIE,C;MACd,OAA4B,OA  
Ab,aAAR,OAAQ,CAAA,eAAI,qDAAJ,C;K;IAOhC,kC;MAkBI,ad3hEO,MAAO,Kc2hEU,gBd3hEV,EcghEH,KA  
W2B,Od3hExB,C;Mc4hEd,WAAW,iBAAa,MAAb,C;MACX,aAAU,CAAV,MAAkB,MAAIB,M;QACI,IAAK,WA  
dqB,GAcP,iCAAK,CAAL,EAdO,EAcE,YAdrB,KAcqB,YAAM,CAAN,EAdF,CACrB,C;MAdT,OAgBO,I;K;wEA  
bX,yB;MAAA,gE;MAAA,oC;MdzHEA,iB;McyhEA,8C;QAQI,ad3hEO,MAAO,Kc2hEK,SAAK,Od3hEV,Ec2hEk  
B,KAAM,Od3hExB,C;Qc4hEd,WAAW,eAAa,MAAb,C;QACX,aAAU,CAAV,MAAkB,MAAIB,M;UACI,IAAK,  
WAAI,UAAU,iCAAK,CAAL,EAAV,EAAM,B,6BAAM,CAAN,EAAnB,CAAJ,C;QAET,OAAO,I;O;KAbX,C;IAg  
BA,kC;MASW,sB;;QAaP,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,qBAAO,W;UAAP,uB;;  
QACd,aAAa,iBAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WajBkB,GAiBJ,iCAAK,KA  
AL,EAjBI,EAiBS,iCAAK,QAAQ,CAAR,IAAL,EAjBT,CaiBIB,C;QAEX,qBAAO,M;;MANBP,yB;K;uFAGJ,yB;  
MAAA,qD;MAAA,gE;MAAA,oC;MAAA,uC;QAUI,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UA  
Ac,OAAO,W;QACrB,aAAa,eAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WAAI,UAAU,i  
CAAK,KAAL,EAAV,EAAuB,iCAAK,QAAQ,CAAR,IAAL,EAAvB,CAAJ,C;QAEX,OAAO,M;O;KAhBX,C;IA  
wBoB,8C;MAAA,mB;QAAE,OAAK,WAAL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OCniE0B,qB  
AAU,C;;MDmiE1D,S;QAAiC,OAAO,W;MACxC,oCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,  
WAAL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OC3iE0B,qBAAU,C;;MD2iE1D,S;QAAiC,OAAO,  
e;MACxC,oCAAgB,8BAAhB,C;K;IEpwEkC,yC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,ChCsLV,K;O;K;LiCtL  
H,wC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,ChC8NV,K;O;K;liC9NC,yC;MAAA,wB;QAAW,OAAA,aAAK,  
KAAL,CjByOV,K;O;K;IkBzOC,0C;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCiMV,K;O;K;4FkC5PzC,qB;MA  
UI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CA  
AJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;M  
AUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,C  
AAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;  
MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,  
CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,q  
B;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBA  
AI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;uGAuC  
X,yB;MA8gHI,8D;MA9gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAqgHT,cAAR,iBAAQ,C;QArgHhB,OAAO,  
OAAcS,sBAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MA0gHI,8D;MA1gHJ,iD;QASe,o  
BAAS,C;QAAT,S;UAAc,gBAigHT,cAAR,iBAAQ,C;QAjgHhB,OAAO,OAAcS,sBAAI,KAJ,CAATC,GAAsD,a  
AAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAsgHI,8D;MATgHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBA6/GT,cAA  
R,iBAAQ,C;QA7/GhB,OAAO,OAAcS,sBAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;M  
AkgHI,8D;MAIlgHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAy/GT,cAAR,iBAAQ,C;QAz/GhB,OAAO,OAAcS,s  
BAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAA  
L,SAAK,EAAU,KAAV,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAAAL,SAAK,EAAU,KAAV,  
C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;iFAYA,gC;MA  
SW,sB;;QA8NS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA9NH,SA8NO,CAAU,OAAV,CA  
AJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MA/NP,yB;K;iFAGJ,gC;MASW,sB;;QA6NS,Q;QAAA,  
2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA7NH,SA6NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;  
YAAP,uB;;QAC9C,qBAAO,I;;MA9NP,yB;K;iFAGJ,gC;MASW,sB;;QA4NS,Q;QAAA,2B;QAAhB,OAAgB,cAA  
hB,C;UAAgB,yB;UAAM,IA5NH,SA4NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAA  
AO,I;;MA7NP,yB;K;iFAGJ,gC;MASW,sB;;QA2NS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAA  
M,IA3NH,SA2NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MA5NP,yB;K;yF

AGJ,yB;MA4nBA,+C;MAkuFI,0D;MA91GJ,uC;QASW,qB;;UA4nBO,Q;UAAA,OAAa,SAytFX,YAAR,iBAAQ,C  
AztFW,CAAb,W;UAAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA9nBc,SA8nBV,CAA  
U,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAhoBP,wB;O;KATJ,C;yFAYA,yB;Mago  
BA,+C;MA0tFI,0D;MA11GJ,uC;QASW,qB;;UAgoBO,Q;UAAA,OAAa,SAitFX,YAAR,iBAAQ,CAjtFW,CAAb,  
W;UAAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAkoBV,CAAU,OAAV,CAAJ  
,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QApoBP,wB;O;KATJ,C;yFAYA,yB;MAooBA,+C;MAktFI,  
0D;MA1tGJ,uC;QASW,qB;;UAooBO,Q;UAAA,OAAa,SAysFX,YAAR,iBAAQ,CAzsFW,CAAb,W;UAAAd,OAAc,  
cAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAtoBc,SAsoBV,CAAU,OAAV,CAAJ,C;cAAwB,oBA  
AO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAxoBP,wB;O;KATJ,C;yFAYA,yB;MAwoBA,+C;MA0sFI,0D;MA11GJ,uC  
;QASW,qB;;UAwoBO,Q;UAAA,OAAa,SAisFX,YAAR,iBAAQ,CAjsFW,CAAb,W;UAAAd,OAAc,cAAd,C;YAAc,  
uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA1oBc,SA0oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB  
;;;UAE5B,oBAAO,I;;;QA5oBP,wB;O;KATJ,C;mFAYA,yB;MAAA,8C;MnCpHA,6B;MmCoHA,4B;QAQI,OnCIH  
mC,cmCkHpB,MAAR,iBAAQ,CnCIHoB,C;O;KmC0GvC,C;mFAWA,yB;MAAA,8C;MnBhHA,+B;MmBgHA,4B;  
QAQI,OnB9GsC,emB8GvB,MAAR,iBAAQ,CnB9GuB,C;O;KmBsG1C,C;mFAWA,yB;MAAA,8C;MpCxLA,+B;  
MoCwLA,4B;QAQI,OpCtLsC,eoCsLvB,MAAR,iBAAQ,CpCtLuB,C;O;KoC8K1C,C;mFAWA,yB;MAAA,8C;MI  
CtLA,iC;MkCsLA,4B;QAQI,OICpLyC,gBkCoL1B,MAAR,iBAAQ,CICpL0B,C;O;KkC4K7C,C;mFAWA,yB;MA  
AA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CA  
AJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;  
QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA  
AO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,  
2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MA  
AM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB  
,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDA  
AvB,C;O;KATV,C;IAYA,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MA  
MI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,G  
AAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGp  
C,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAA  
wB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q  
AAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;  
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAA  
O,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,C  
AAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;2FAGX,yB;MAkqGI,8D;MALqGJ,iD;QAQoE,oBAAS,C;QAAT,S;U  
AAc,gBA2pGT,cAAR,iBAAQ,C;;QA3pGhB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;K  
APjE,C;2FAUA,yB;MAGqGI,8D;MAhGJ,iD;QAQoE,oBAAS,C;QAAT,S;UAAc,gBAypGT,cAAR,iBAAQ,C;;QA  
z  
pGhB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA8pGI,8D;MA9p  
GJ,iD;QAQoE,oBAAS,C;QAAT,S;UAAc,gBAupGT,cAAR,iBAAQ,C;;QAvpGhB,OAAO,OAAc,sBAAI,KAAJ,C  
AAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA4pGI,8D;MA5pGJ,iD;QAQoE,oBAAS,C;QAAT,S;UAA  
c,gBAqpGT,cAAR,iBAAQ,C;;QArpGhB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE  
,C;IAUA,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBAknGT,gBAAR,iBAAQ,C;;MALnGhB,OAAO,OAAc,sBAAI,  
KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBA+mGT,gBAAR,iBAAQ,C;;MA/mG  
hB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBA4mGT,g  
BAAR,iBAAQ,C;;MA5mGhB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;M  
AAT,S;QAAC,gBAymGT,gBAAR,iBAAQ,C;;MAzmGhB,OAAO,OAAc,sBAAI,KAAJ,CAAtC,GAAsD,I;K;uFA  
GjE,yB;MAAA,kD;MAAA,qC;QAQI,OAAe,QAAR,iBAAQ,EAAQ,OnBrdY,KmBqdpB,C;O;KAPnB,C;uFAUA,yB  
;MAAA,kD;MAAA,qC;QAQI,OAAe,QAAR,iBAAQ,EAAQ,OpClhBY,KoCkhBpB,C;O;KAPnB,C;uFAUA,yB;MAAA,k  
D;MAAA,qC;QAQI,OAAe,QAAR,iBAAQ,EAAQ,OICjhBc,KkCihBtB,C;O;KAPnB,C;iGAUA,yB;MAAA,sC;Mn  
C5ZA,6B;MmC4ZA,0BAOGC,yB;QnCnahC,6B;emCmagC,6B;UAAA,qB;YAAE,yBnCzZK,cmCyZK,EnCzZL,C

mCyZL,C;W;S;OAAF,C;MAPhC,uC;QAOMB,kBAAR,iB;QAAQ,uB;;UtC40Bf,0D;YACI,IsC70B0B,UnCzZK,cH  
suCjB,YAAK,KAAL,CGtuCiB,CmCyZL,CtC60B1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCj1BP,  
0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MnBvZA,+B;MmBuZA,0BAOgC,yB;QnB9ZhC,+B;emB8ZgC,6B;UAAA,q  
B;YAAE,yBnBpZQ,emBoZE,EnBpZF,CmBoZR,C;W;S;OAAF,C;MAPhC,uC;QAOMB,kBAAR,iB;QAAQ,uB;;Ut  
C80Bf,0D;YACI,IsC/0B0B,UnBpZQ,enBmuCpB,YAAK,KAAL,CmBnuCoB,CmBoZR,CtC+0B1B,C;cACI,sBAA  
O,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCn1BP,0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MpC9dA,+B;MoC8dA,0BA  
OgC,yB;QpCrehC,+B;eoCqegC,6B;UAAA,qB;YAAE,yBpC3dQ,eoC2dE,EpC3dF,CoC2dR,C;W;S;OAAF,C;MAP  
hC,uC;QAOMB,kBAAR,iB;QAAQ,uB;;UtCgyBf,0D;YACI,IsCjyB0B,UpC3dQ,eF4vCpB,YAAK,KAAL,CE5vCo  
B,CoC2dR,CtCiyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCryBP,0B;O;KAPJ,C;iGAUA,yB;MA  
AA,sC;MIC3dA,iC;MkC2dA,0BAOgC,yB;QlClehC,iC;ekCkegC,6B;UAAA,qB;YAAE,yBICxdW,gBkCwdD,ElC  
dC,CkCwdX,C;W;S;OAAF,C;MAPhC,uC;QAOMB,kBAAR,iB;QAAQ,uB;;UtCkyBf,0D;YACI,IsCnyB0B,UlCxd  
W,gBJ2vCvB,YAAK,KAAL,Ci3vCuB,CkCwdX,CtCmyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;Q  
sCvyBP,0B;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCm5BA,0D;MAAA,+C;MGv1CA,6B;MmCocA,yBAO+B,yB;Q  
nC3c/B,6B;emC2c+B,6B;UAAA,qB;YAAE,yBnBcjM,cmCicI,EnCjeJ,CmCicN,C;W;S;OAAF,C;MAP/B,uC;QA  
OMB,kBAAR,iB;QAAQ,sB;;UtCg5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,u  
B;YACV,IsCj5ByB,UnCjcM,cHk1CjB,YAAK,KAAL,CGl1CiB,CmCicN,CtCi5BzB,C;cACI,qBAAO,K;cAAP,uB;  
;UAGR,qBAAO,E;;;QsCr5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCq5BA,0D;MAAA,+C;MmBp1CA,+B;M  
mB+bA,yBAO+B,yB;QnBtc/B,+B;emBsc+B,6B;UAAA,qB;YAAE,yBnB5bS,emB4bC,EnB5bD,CmB4bT,C;W;S;  
OAAF,C;MAP/B,uC;QAOMB,kBAAR,iB;QAAQ,sB;;UtCk5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UA  
Ad,OAAc,cAAAd,C;YAAc,uB;YACV,IsCn5ByB,UnB5bS,enB+0CpB,YAAK,KAAL,CmB/0CoB,CmB4bT,CtCm5  
BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QsCv5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCu2B  
A,0D;MAAA,+C;ME72CA,+B;MoCsgBA,yBAO+B,yB;QpC7gB/B,+B;eoC6gB+B,6B;UAAA,qB;YAAE,yBpCng  
BS,eoCmgBC,EpCngBD,CoCmgBT,C;W;S;OAAF,C;MAP/B,uC;QAOMB,kBAAR,iB;QAAQ,sB;;UtCo2BD,Q;U  
AAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IsCr2ByB,UpCngBS,eFw2CpB,  
YAAK,KAAL,CEx2CoB,CoCmgBT,CtCq2BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QsCz2BP,yB;  
O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCy2BA,0D;MAAA,+C;MI52CA,iC;MkCmgBA,yBAO+B,yB;QIC1gB/B,iC;  
kC0gB+B,6B;UAAA,qB;YAAE,yBIChgBY,gBkCggBF,ElChgBE,CkCggBZ,C;W;S;OAAF,C;MAP/B,uC;QAOMB  
,kBAAR,iB;QAAQ,sB;;UtCs2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;Y  
ACV,IsCv2ByB,UlChgBY,gBJu2CvB,YAAK,KAAL,CiV2CuB,CkCggBZ,CtCu2BzB,C;cACI,qBAAO,K;cAAP,u  
B;;;UAGR,qBAAO,E;;;QsC32BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,4C;MnC5eA,6B;MmC4eA,4B;QAWI,OnC7e  
mC,cmC6epB,KAAR,iBAAQ,CnC7eoB,C;O;KmCkevC,C;iFAcA,yB;MAAA,4C;MnB3eA,+B;MmB2eA,4B;QAW  
I,OnB5esC,emB4evB,KAAR,iBAAQ,CnB5euB,C;O;KmBie1C,C;iFAcA,yB;MAAA,4C;MpCtjBA,+B;MoCsjBA,4  
B;QAWI,OpCvjBsC,eoCujBvB,KAAR,iBAAQ,CpCvjBuB,C;O;KoC4iB1C,C;iFAcA,yB;MAAA,4C;MICvjBA,iC;  
MkCujBA,4B;QAWI,OICxjByC,gBkCwjB1B,KAAR,iBAAQ,CiCxB0B,C;O;KkC6iB7C,C;iFAcA,yB;MAAA,+C;  
MAAA,iE;MA83FI,0D;MA93FJ,uC;QAWkB,Q;QAAA,OAAa,SAm3FX,YAn3FF,SAm3FN,QAAQ,CAAn3FW,CA  
Ab,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YA  
AwB,OAAO,O;;QAEnc,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;iFAkBA,yB;MAAA,+C;MAAA,iE;MAo3FI,0  
D;MAp3FJ,uC;QAWkB,Q;QAAA,OAAa,SAy2FX,YAz2FF,SAy2FN,QAAQ,CAz2FW,CAAb,W;QAAd,OAAc,cA  
Ad,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnc,  
MAAM,gCAAuB,mDAAvB,C;O;KafV,C;iFAkBA,yB;MAAA,+C;MAAA,iE;MA02FI,0D;MA12FJ,uC;QAWkB,  
Q;QAAA,OAAa,SA+1FX,YA/1FF,SA+1FN,QAAQ,CA/1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,c  
AAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnc,MAAM,gCAAuB,mDA  
AvB,C;O;KafV,C;iFAkBA,yB;MAAA,+C;MAAA,iE;MAg2FI,0D;MAh2FJ,uC;QAWkB,Q;QAAA,OAAa,SAq1F  
X,YAr1FF,SAq1FN,QAAQ,CAr1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;  
UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnc,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;+FAk  
BA,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OnC9sBM,KmC8sBIB,C;O;KAPnB,C;+FAU  
A,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OpC1wBQ,KoC0wBpB,C;O;KAPnB,C;+FAUA,y





I,IAAJ,C;YAACL,WAAW,I;;;QAE nB,OAAO,I;O;KAIBX,C;qFAqBA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,g  
B;QAgRA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAhRa,SAgRT,CAAU,OAAV,CAAJ,C;Y  
AAwB,WAAY,WAAL,OAAJ,C;;QAhR1D,OAIRO,W;O;KA1RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kB  
AAS,gB;QAIrA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAjRc,SAiRV,CAAU,OAAV,CAA  
J,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAJR1D,OAKRO,W;O;KA3RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QAS  
W,kBAAS,gB;QAKRA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAIRc,SAkRV,CAAU,OAA  
V,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAIr1D,OAmRO,W;O;KA5RX,C;qFAYA,yB;MAAA,+D;MAAA,  
uC;QASW,kBAAS,gB;QAmRA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IANRe,SAmRX,CA  
AU,OAAV,CAAJ,C;YAAwB,WAAY,WAAL,OAAJ,C;;QAnR1D,OAoRO,W;O;KA7RX,C;kGAYA,yB;MAAA,+D  
;MAAA,uC;QAWW,kBAAGB,gB;QAm5HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA  
11HT,IAzDsC,SAyDIC,EA01HkB,cA11HIB,EA01HkB,sBA11HIB,WA01H2B,IA11H3B,CAAJ,C;YAA2C,sBA01  
HZ,IA11HY,C;;QAzD/C,OA2DO,W;O;KAtEX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAK5  
HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAt1HT,IA5DuC,SA4DnC,EAs1HkB,cAt1H  
lB,EAs1HkB,sBA11HIB,WA1H2B,IA11H3B,CAAJ,C;YAA2C,sBA11HZ,IA11HY,C;;QA5D/C,OA8DO,W;O;KAZ  
EX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAI5HV,gB;QADb,YAAY,C;QACC,2B;QAAb,O  
AAa,cAAb,C;UAAa,sB;UAl1HT,IA/DuC,SA+DnC,EAK1HkB,cAl1HIB,EAK1HkB,sBA11HIB,WAK1H2B,IA11H3  
B,CAAJ,C;YAA2C,sBAk1HZ,IA11HY,C;;QA/D/C,OAIEO,W;O;KA5EX,C;mGAcA,yB;MAAA,+D;MAAA,uC;Q  
AWW,kBAAGB,gB;QAg5HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA90HT,IAIEwC,  
SAkEpC,EA80HkB,cA90HIB,EA80HkB,sBA90HIB,WA80H2B,IA90H3B,CAAJ,C;YAA2C,sBA80HZ,IA90HY,C  
;;QAIE/C,OAoEO,W;O;KA/EX,C;uGAcA,6C;MAS2HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;  
QAAa,sB;QA11HT,IAAI,WA01HkB,cA11HIB,EA01HkB,sBA11HIB,WA01H2B,IA11H3B,CAAJ,C;UAA2C,sBA  
01HZ,IA11HY,C;;MAE/C,OAAO,W;K;uGAGX,6C;MAK2HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cA  
Ab,C;QAAa,sB;QAt1HT,IAAI,WA1HkB,cAt1HIB,EAs1HkB,sBA11HIB,WA1H2B,IA11H3B,CAAJ,C;UAA2C,s  
BA11HZ,IA11HY,C;;MAE/C,OAAO,W;K;uGAGX,6C;MA81HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,c  
AAb,C;QAAa,sB;QAI1HT,IAAI,WAK1HkB,cAl1HIB,EAK1HkB,sBA11HIB,WAK1H2B,IA11H3B,CAAJ,C;UAA2  
C,sBAk1HZ,IA11HY,C;;MAE/C,OAAO,W;K;uGAGX,6C;MA01HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OA  
Aa,cAAb,C;QAAa,sB;QA90HT,IAAI,WA80HkB,cA90HIB,EA80HkB,sBA90HIB,WA80H2B,IA90H3B,CAAJ,C;  
UAA2C,sBA80HZ,IA90HY,C;;MAE/C,OAAO,W;K;2FAGX,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,gB;QAg  
DH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAhDY,SAgDX,CAAU,OAAV,CAAL,C;  
YAAyB,WAAY,WAAL,OAAJ,C;;QAhD3D,OAI DO,W;O;KA1DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;QASW,k  
BAAY,gB;QAI DH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAjDa,SAiDZ,CAAU,OA  
AV,CAAL,C;YAAyB,WAAY,WAAL,OAAJ,C;;QAJD3D,OAKDO,W;O;KA3DX,C;2FAYA,yB;MAAA,+D;MAAA,  
uC;QASW,kBAAY,gB;QAKDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAIDa,SAk  
DZ,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAL,OAAJ,C;;QAID3D,OAmDO,W;O;KA5DX,C;2FAYA,yB;M  
AAA,+D;MAAA,uC;QASW,kBAAY,gB;QAmDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,I  
AAI,CAnDc,SAmDb,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAL,OAAJ,C;;QAnD3D,OAoDO,W;O;KA7DX,  
C;+FAYA,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV  
,CAAL,C;UAAyB,WAAY,WAAL,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OA  
AgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAL,OAAJ,C;;MAC  
3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,  
UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAL,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2  
B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAL,  
OAAJ,C;;MAC3D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAA  
M,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAL,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;  
MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAA  
L,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAA  
M,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAL,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;  
MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAA



I,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OhCvjEe,W;;MgCwjEtC,OAA4D,SA0iDrD,cAAkB,cAAR,iBAAQ,EA1iDN,OAAQ,MA0iDF,EA1iDS,OAAQ,aAAR,GAAuB,CAAvB,IA0iDT,C AAIb,CA1iDqD,C;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OhCjkEe,W;;MgCkkEtC,OAA4D,SAgjDr D,eAAmB,cAAR,iBAAQ,EAhjDP,OAAQ,MAgjDD,EAhjDQ,OAAQ,aAAR,GAAuB,CAAvB,IAgjDR,CAAnB,C AhjDqD,C;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OhC3kEe,W;;MgC4kEtC,OAA4D,UAsjDrD,eAA mB,cAAR,iBAAQ,EA1jDP,OAAQ,MA1jDD,EA1jDQ,OAAQ,aAAR,GAAuB,CAAvB,IA1jDR,CAAnB,CAtjDqD,C ;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OhCrlEe,W;;MgCslEtC,OAA4D,UA4jDrD,gBAAoB,cAAR,i BAAQ,EA5jDR,OAAQ,MA4jDA,EA5jDO,OAAQ,aAAR,GAAuB,CAAvB,IA4jDP,CAApB,CA5jDqD,C;K;IAGh E,sC;MASKB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OA AO,W;MACtB,WAAW,iBAAGB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAAC,uB;QACV,IAAK,WAAI,sB AAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MASKB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB, C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAc,c AAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MASKB,Q;MAHd,W AAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iB AAIb,IAAjB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET ,OAAO,I;K;IAGX,sC;MASKB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CA AZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAkB,IAAIb,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAAC,uB;QACV,I AAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,2C;MAMI,OAAO,cAAkB,aAAR,iBAAQ,EAAW, OAAx,CAAIb,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MA MI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,aAAR,iBAAQ ,EAAW,OAAx,CAApB,C;K;IAGX,2C;MAMI,OAAO,cAAkB,cAAR,iBAAQ,EAAW,OAAx,CAAIb,C;K;IAGX, 2C;MAMI,OAAO,eAAmB,cAAR,iBAAQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,i BAAQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,cAAR,iBAAQ,EAAW,OAAx,CAApB,C; K;IAGX,+B;MAGBiB,Q;MxB7xEb,IAAI,EwBuxEI,KAAK,CxBvxET,CAAJ,C;QACI,cwBsxEc,sD;QxBrxEd,MA AM,gCAAYB,OAAQ,WAAjC,C;;MwBsxEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAA T,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,Y AAY,C;MACZ,WAAW,iBAAGB,CAAhB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IA AJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxB7yEb,IAAI,EwB+yE I,KAAK,CxB/yET,CAAJ,C;QACI,cwB8yEc,sD;QxB7yEd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB8yEV,IAAI, MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAA T,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE ,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAE R,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxB70Eb,IAAI,EwBu0EI,KAAK,CxBv0ET,CAAJ,C;QACI,cwBs0Ec,sD;Qx Br0Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBs0EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI, KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP, C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAA K,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxB2Eb,IA AI,EwB+1EI,KAAK,CxB/1ET,CAAJ,C;QACI,cwB81Ec,sD;QxB71Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB 81EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI, MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAkB,CA AIb,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C; UACI,K;;MAER,OAAO,I;K;IAGX,mC;MxB72EI,IAAI,EwBu3EI,KAAK,CxBv3ET,CAAJ,C;QACI,cwBs3Ec,sD; QxB3Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBs3EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WA AW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sB AAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAGB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,U AA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxB14EI,IAAI,EwB 44EI,KAAK,CxB54ET,CAAJ,C;QACI,cwB24Ec,sD;QxB14Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB24EV,I AAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MA

CtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAA  
iB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;  
MACT,OAAO,I;K;IAGX,mC;MxBv5EI,IAAI,EwBi6EI,KAAK,CxBj6ET,CAAJ,C;QACI,cwBg6Ec,sD;QxB/5Ed,  
MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBg6EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;M  
ACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OA  
AO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA  
7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxB56EI,IAAI,EwBs7EI,KAA  
K,CxBt7ET,CAAJ,C;QACI,cwBq7Ec,sD;QxBp7Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBq7EV,IAAI,MAA  
K,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,  
MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAkB,CAAI  
B,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,  
OAAO,I;K;mGAGX,yB;MAAA,4C;MAAA,gD;MA s2CI,8D;Mat2CJ,uC;QASI,iBA61CgB,cAAR,iBAAQ,CA71C  
hB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ  
,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAIbA,yB;MAAA,4C;MAAA,gD;MA61CI,8D;MA71CJ,uC;QA  
SI,iBAo1CgB,cAAR,iBAAQ,Cap1ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,C  
AAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAIbA,yB;MAAA,4C;M  
AAA,gD;MAo1CI,8D;MAp1CJ,uC;QASI,iBA20CgB,cAAR,iBAAQ,CA30ChB,WAA+B,CAA/B,U;UACI,IAAI,C  
AAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O  
;KAdX,C;mGAIbA,yB;MAAA,4C;MAAA,gD;MA20CI,8D;MA30CJ,uC;QASI,iBAk0CgB,cAAR,iBAAQ,CA10C  
hB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ  
,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;  
QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,W  
AAI,IAAJ,C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QAC  
E,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,I  
AAJ,C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;  
QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,  
C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAA  
b,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QA  
ET,OAAO,I;O;KafX,C;uFAkBA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MA  
AA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAA  
Q,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;  
MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;uFAgBA,yB;MAAA,kD;MAAA,gD;QAa  
Y,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;SFAgBA,yB;MAAA,kD;MAAA,gD;QAaY,QAA  
R,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;IAGBA,gC;MAMI,IAAI,mBAAJ,C;QA Ae,OAAO,W;MA  
CtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QA Ae,OAAO,W;  
MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QA Ae,OAAO,W;  
MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QA Ae,OA  
AO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;kGAGX,yB;MAAA,8D;MAAA,uC;MAA  
A,4B;QAOI,OAAO,mBAAkB,cAAR,iBAAQ,CAAI B,C;O;KAPX,C;kGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4  
B;QAOI,OAAO,oBAAMB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4B;  
QAOI,OAAO,oBAAMB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,2C;MAAA,4B;Q  
AOI,OAAO,qBAAoB,cAAR,iBAAQ,CAApB,C;O;KAPX,C;IAUA,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MA  
MI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,uC;  
MAQI,aA8+BgB,gBAAR,iBAAQ,CA9+BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAA  
R,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CA  
AL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aAs+BgB,gBAAR,iBAAQ,CAt+BhB,OAA2B,CAA3B,M;QACI,QAAQ,M  
AAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAA

L,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA89BgB,gBAAR,iBAAQ,CA99BhB,OA  
A2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBA  
AK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aAs9BgB,  
gBAAR,iBAAQ,CAt9BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW  
,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C  
;;K;IAIR,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QApSI,UAAR,iBAAQ,C;;K;IAySZ,sC;MAMI,IAAI,iBAAO  
,CAAX,C;QACI,iB;QAtSI,UAAR,iBAAQ,C;;K;IA2SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QAxSI,UAAR  
,iBAAQ,C;;K;IA6SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QA1SI,UAAR,iBAAQ,C;;K;IA+SZ,6B;MAMoB  
,kBA+nBT,cAAU,iBvB58EO,QuB48EjB,C;MA/nBiB,mB;MAAxB,OAAiC,SrBv3F1B,WqBu3F0B,C;K;IAGrC,8B  
;MAMoB,kBAkoBT,eAAmB,UAAR,iBAAQ,CAAnB,C;MAl0BiB,mB;MAAxB,OAAiC,SrBh4F1B,WqBg4F0B,C;  
K;IAGrC,8B;MAMoB,kBAqoBT,eAAW,iBvBx/EM,QuBw/EjB,C;MAroBiB,mB;MAAxB,OAAiC,UrBz4F1B,Wq  
By4F0B,C;K;IAGrC,8B;MAMoB,kBAwoBT,gBAAY,iBvB1/EK,QuB0/EjB,C;MAxoBiB,mB;MAAxB,OAAiC,Ur  
B15F1B,WqBk5F0B,C;K;IAGrC,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA01Bd,cA11BA,SA01BU,  
QvB58EO,QuB48EjB,C;MA11BsB,mB;MAA7B,OrB55FO,W;K;IqB+5FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OA  
AO,S;MACD,kBA41Bd,eAAmB,UA51BnB,SA41BW,QAAQ,CAAnB,C;MA51BsB,mB;MAA7B,OrBt6FO,W;K;Iq  
By6FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA81Bd,eA91BA,SA81BW,QvBx/EM,QuBw/EjB,C;  
MA91BsB,mB;MAA7B,OrBh7FO,W;K;IqBm7FX,mC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAgmbd  
,gBAhmBA,SagmBY,QvB1/EK,QuB0/EjB,C;MAhmBsB,mB;MAA7B,OrB17FO,W;K;IqB67FX,4C;MAMI,IAAI,  
mBAAJ,C;QAAe,OAAO,S;MACD,kBAkjBd,cAljBA,SAkjBU,QvB58EO,QuB48EjB,C;MALjBsB,8B;MAA7B,Or  
Bp8FO,W;K;IqBu8FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAojBd,eAAmB,UApjBnB,SAojBW,  
QAAQ,CAAnB,C;MApjBsB,8B;MAA7B,OrB98FO,W;K;IqBi9FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;M  
ACD,kBASjBd,eAtjBA,SAsjBW,QvBx/EM,QuBw/EjB,C;MATjBsB,8B;MAA7B,OrBx9FO,W;K;IqB29FX,6C;MA  
MI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAwjBd,gBAxjBA,SAwjBY,QvB1/EK,QuB0/EjB,C;MAXjBsB,8B;  
MAA7B,OrBl+FO,W;K;IqBq+FX,uC;MAQoB,kBAygbT,cAAU,iBvB58EO,QuB48EjB,C;MAzgBiB,mB;MAAxB  
,OAAiC,YrB7+F1B,WqB6+F0B,C;K;IAGrC,wC;MAQoB,kBA0gBT,eAAmB,UAAR,iBAAQ,CAAnB,C;MA1gBi  
B,mB;MAAxB,OAAiC,YrBx/F1B,WqBw/F0B,C;K;IAGrC,wC;MAQoB,kBA2gBT,eAAW,iBvBx/EM,QuBw/EjB,  
C;MA3gBiB,mB;MAAxB,OAAiC,YrBngG1B,WqBmgG0B,C;K;IAGrC,wC;MAQoB,kBA4gBT,gBAAY,iBvB1/E  
K,QuB0/EjB,C;MA5gBiB,mB;MAAxB,OAAiC,YrB9gG1B,WqB8gG0B,C;K;4FAGrC,qB;MAQI,OAAO,iB;K;0F  
AGX,qB;MAQI,OAAO,iB;K;4FA+BX,qB;MAQI,OAAO,iB;K;8FAGX,qB;MAQI,OAAO,iB;K;8FAGX,yB;MAA  
A,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;4FAWA,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,  
mBAAU,SAAV,C;O;KARX,C;8FAWA,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;g  
GAWA,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAY,SAAZ,C;O;KARX,C;IAWA,2C;MASI,OAAAY,gBAAL,  
SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAAY,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAAY,g  
BAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAAY,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MAOI,O  
AAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;  
MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IA  
GzB,sC;MAQI,OAAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAA  
Y,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAMI,OAAqB,gBAAd,4CAAc,  
C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;  
MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAUI,OAAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAAY,kBA  
AL,SAAK,C;K;IAGhB,sC;MAUI,OAAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAAY,kBAAL,SAAK,C;K;IAGh  
B,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MA  
QW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;M  
AAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IFAGjD,yB;MvBxhFA,8C;MuBwhFA,kF;  
QAmB6D,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;QvBvhF1H,UuBwhF  
A,iBvBxhFA,EuBwhFiB,WAAAY,QvBxhF7B,EuBwhFsC,iBvBxhFtC,EuBwhFyD,UvBxhFzD,EuBwhFqE,QvBxhF  
rE,C;QuByhFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvBxhFA,8C;MuBwhFA,kF;QAmB+D,iC;UAAA,oBAAY

B,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;QvBvhF5H,UuBwhFA,iBvBxhFA,EuBwhFiB,W  
AAY,QvBxhF7B,EuBwhFsC,iBvBxhFtC,EuBwhFyD,UvBxhFzD,EuBwhFqE,QvBxhFrE,C;QuByhFA,OAAO,W;  
O;KArBX,C;wFAwBA,yB;MvBxnFA,8C;MuBwnFA,kF;QAmB+D,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAA  
kB,C;QAAG,wB;UAAA,WAAgB,c;QvBvnF5H,UuBwnFA,iBvBxnFA,EuBwnFiB,WAAy,QvBxnF7B,EuBwnFsC  
,iBvBxnFtC,EuBwnFyD,UvBxnFzD,EuBwnFqE,QvBxnFrE,C;QuBynFA,OAAO,W;O;KArBX,C;wFAwBA,yB;M  
vBxnFA,8C;MuBwnFA,kF;QAmBiE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WA  
AgB,c;QvBvnF9H,UuBwnFA,iBvBxnFA,EuBwnFiB,WAAy,QvBxnF7B,EuBwnFsC,iBvBxnFtC,EuBwnFyD,UvB  
xnFzD,EuBwnFqE,QvBxnFrE,C;QuBynFA,OAAO,W;O;KArBX,C;kFAwBA,yB;MAAA,uC;MAAA,4B;QASI,OA  
AO,mBAAU,iBvB58EO,QuB48EjB,C;O;KATX,C;oFAYA,yB;MAAA,gD;MAAA,yC;MAAA,4B;QASI,OAAO,o  
BAAmB,OAAR,iBAAQ,CAAnB,C;O;KATX,C;oFAYA,yB;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAW,iBvBx/  
EM,QuBw/EjB,C;O;KATX,C;oFAYA,yB;MAAA,2C;MAAA,4B;QASI,OAAO,qBAAY,iBvB1/EK,QuB0/EjB,C;O  
;KATX,C;oFAYA,yB;MAAA,gD;MAAA,uC;MAAA,qC;QAWI,OAAO,mBAAkB,OAAR,iBAAQ,EAAO,OAAP,  
CAAIB,C;O;KAXX,C;oFAcA,yB;MAAA,gD;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EA  
AO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;MAAA,+C;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,  
iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;MAAA,gD;MAAA,2C;MAAA,qC;QAWI,OAAO,qBAA  
oB,OAAR,iBAAQ,EAAO,OAAP,CAApB,C;O;KAXX,C;4FAcA,yB;MAAA,0D;MAAA,uC;MAAA,gD;QAaI,OA  
AO,mBAAkB,YAAR,iBAAQ,EAAY,SAAZ,EAAuB,OAAvB,CAAIB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MA  
AA,yC;MAAA,gD;QAaI,OAAO,oBAAmB,YAAR,iBAAQ,EAAY,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C;  
8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD;QAaI,OAAO,oBAAmB,YAAR,iBAAQ,EAAY,SAAZ,EAAuB,O  
AAvB,CAAnB,C;O;KAbX,C;6FAgBA,yB;MAAA,0D;MAAA,2C;MAAA,gD;QAaI,OAAO,qBAAoB,YAAR,iBA  
AQ,EAAY,SAAZ,EAAuB,OAAvB,CAApB,C;O;KAbX,C;IAGBA,sD;MAWYc,yB;QAAA,YAAiB,C;MAAG,uB;  
QAAA,UAAe,c;MACIE,OAAR,iBAAQ,EAAC,OnCv8GoB,KmCu8GzB,EAAAsB,SAAtB,EAaiC,OAAjC,C;K;IA  
GZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACIE,OAAR,iBAAQ,EAAC,OnB38GsB,K  
mB28G3B,EAAuB,SAAvB,EAakC,OAAIC,C;K;IAGZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,U  
AAe,c;MACIE,OAAR,iBAAQ,EAAC,OpC7gHsB,KoC6gH3B,EAAuB,SAAvB,EAakC,OAAIC,C;K;IAGZ,wD;M  
AW6C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACpE,OAAR,iBAAQ,EAAC,OICjhHwB,KkCihH7B,E  
AAwB,SAAxB,EAAMC,OAAnc,C;K;8FASR,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KA  
AhB,C;8FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;  
MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YA  
AR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;kG  
AQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4  
B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAA  
Q,C;O;KAAhB,C;iFAEJ,yB;MAAA,uC;MvBvoEA,iD;MuBuoEA,qC;QAOqB,4B;QAAA,gBAAU,OnC9jHM,K;Q  
mC8jHjC,OAAO,mBvBzoEA,2BAxIK,gBAAW,SAAX,EAwIL,CuByoEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;M  
vBzoEA,iD;MuByoEA,qC;QAOI,OAAO,oBvB3oEA,qBuB2oEW,iBvB3oEX,EAXIK,mBuBmxEGb,OnB7jHO,KJ0  
yCvB,CAwIL,CuB2oEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvB3qEA,iD;MuB2qEA,qC;QAOsB,4B;QAAA,g  
BAAU,OpC1nHO,K;QoC0nHnC,OAAO,oBvB7qEA,2BAxIK,eAAY,SAAZ,EAwIL,CuB6qEA,C;O;KAPX,C;iFA  
UA,yB;MAAA,2C;MvB7qEA,iD;MuB6qEA,qC;QAOuB,4B;QAAA,gBAAU,OICznHQ,K;QkCynHrC,OAAO,qBv  
B/qEA,2BAxIK,gBAAa,SAAb,EAwIL,CuB+qEA,C;O;KAPX,C;IAUA,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;  
MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;Q  
AAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnCvmHX,K;;MmCwmHjC,OAAO,cAAU,MAAV,C;K;I  
AGX,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IA  
AP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnBxmH  
T,K;;MmBymHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,U  
AAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,  
OAAO,cAAP,EAAO,sBAAP,YAAkB,OpCvqHT,K;;MoCwqHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB,  
UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;M  
AAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OICxqHP,K;;MkCyqHrC,OA

AO,gBAAy,MAAZ,C;K;iFAGX,yB;MAAA,uC;MvB/tEA,iD;MuB+tEA,sC;QAOI,OAAO,mBvBjuEA,qBuBiuEU,  
iBvBjuEV,EUbiuEoB,QAAS,QvBjuE7B,CuBiuEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBjuEA,iD;MuBiuEA,s  
C;QAOI,OAAO,oBvBnuEA,qBuBmuEW,iBvBnuEX,EUbmuEqB,QAAS,QvBnuE9B,CuBmuEA,C;O;KAPX,C;iF  
AUA,yB;MAAA,yC;MvBnwEA,iD;MuBmwEA,sC;QAOI,OAAO,oBvBrwEA,qBuBqwEW,iBvBrwEX,EUbqwEq  
B,QAAS,QvBrwE9B,CuBqwEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBrwEA,iD;MuBqwEA,sC;QAOI,OAAO  
,qBvBvwEA,qBuBuwEY,iBvBvwEZ,EUbuwEsB,QAAS,QvBvwE/B,CuBuwEA,C;O;KAPX,C;IAUA,2B;MAQI,I  
AAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMb,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAA  
O,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMb,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C  
;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMb,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YA  
AU,SAAV,EAAGB,CAAhB,EAAMb,cAAnB,C;K;IAGIB,+C;MAA0B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,U  
AAe,c;MACzD,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAh  
B,EAA2B,OAA3B,C;K;IAGJ,+C;MAA2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAA,2BA  
AkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAG  
J,+C;MAA2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7  
B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAA4B,yB;QAAA,YA  
AiB,C;MAAG,uB;QAAA,UAAe,c;MAC3D,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb,  
YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C;MAh8C  
Q,WAAR,iBAAQ,EAi8CA,SAj8CA,EAi8CW,OAJ8CX,C;K;IAo8CZ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C  
;MAj8CQ,WAAR,iBAAQ,EAk8CA,SAI8CA,EAk8CW,OAI8CX,C;K;IAq8CZ,0D;MAaI,kBAAK,SAAL,EAAGB,  
OAAhB,C;MAI8CQ,UAAAR,iBAAQ,EA8CA,SA8CA,EA8CW,OAN8CX,C;K;IAS8CZ,0D;MAaI,kBAAK,SA  
AL,EAAGB,OAAhB,C;MAN8CQ,WAAR,iBAAQ,EAo8CA,SAp8CA,EAo8CW,OAP8CX,C;K;8FAu8CZ,qB;MAQ  
I,OAAO,iBvB3jGiB,Q;K;4FuB8jG5B,qB;MAQI,OAAO,iBvBljGiB,Q;K;8FuBqjG5B,yB;MAAA,gD;MAAA,4B;Q  
AQI,OAAe,OAAR,iBAAQ,C;O;KARnB,C;gGAWA,qB;MAQI,OAAO,iBvBljGiB,Q;K;IuB2lIGL,gD;MAAA,wB;Q  
AAW,qCAAK,KAAL,C;O;K;IANIC,iC;MAMI,OAAO,iBAAM,cAAN,EAAY,8BAAZ,C;K;IASY,kD;MAAA,wB;  
QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA,w  
B;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA  
,wB;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASiB,gD;MA  
AA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OJnqIO,eAAW,+BImqIA,gBJnqIA,GAAgB,kBImqIV,8  
BJnqIU,CAAhB,CAAX,C;K;gGIsqIX,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SvBppGM,QuBopGjB,C;  
O;KARX,C;IAiB2B,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IANtC,gC;MAMI,OHvrIO,cAAU,gCGurIA,g  
BHvrIA,GAAe,iBGurIT,6BHvrIS,CAAF,CAAV,C;K;8FG0rIX,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,  
SvBppGO,QuBopGjB,C;O;KARX,C;IAiB4B,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OF  
3sIO,eAAW,kBE2sIA,gBF3sIA,EAAGB,kBE2sIV,8BF3sIU,CAAhB,CAAX,C;K;gGE8sIX,yB;MAAA,gD;MAAA,  
yC;MAAA,4B;QAQI,OAAO,oBAAGB,OAAL,SAAK,CAAhB,C;O;KARX,C;IAiB6B,kD;MAAA,wB;QAAW,0BA  
AK,KAAL,C;O;K;IANxC,kC;MAMI,OD/tIO,gBAAy,gCC+tIA,gBD/tIA,GAAiB,mBC+tIX,+BD/tIW,CAAjB,CA  
AZ,C;K;kGCKuIX,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAy,SvBtsGK,QuBssGjB,C;O;KARX,C;mGAWA  
,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAAyC,cAAIB,YAAY,cAAZ,CAAKB,EAAC,EA  
Ad,CAAzC,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,a  
AsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAaUB,M;O;Kaf3B,C;mGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;  
MAAA,2C;QAcI,aAAa,mBAA0C,cAAIB,YAAY,cAAZ,CAAKB,EAAC,EAAd,CAA1C,C;QAsEG,Q;QAAA,2B;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEh  
B,OAaUB,M;O;Kaf3B,C;kGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA0C,cAA  
IB,YAAY,cAAZ,CAAKB,EAAC,EAAd,CAA1C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;  
UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAaUB,M;O;Kaf3B,C;mGakBA,y  
B;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA2C,cAAIB,YAAY,cAAZ,CAAKB,EAAC,EA  
Ad,CAA3C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,a  
AsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAaUB,M;O;Kaf3B,C;uGakBA,iD;MAyOB,Q;MAAA,2B;MAAhB,OA  
gB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAG

X,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uFAGX,yB;MAAA,+D;MAoLA,gD;MApLA,uC;QASW,kBAAU,gB;QAKLD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAnL6B,SAmLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QApLhB,OAsLO,W;O;KA/LX,C;uFAYA,yB;MAAA,+D;MAsLA,gD;MatLA,uC;QASW,kBAAU,gB;QAoLD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WArL6B,SAqLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtLhB,OAwLO,W;O;KAjMX,C;uFAYA,yB;MAAA,+D;MAwLA,gD;MAxLA,uC;QASW,kBAAU,gB;QAsLD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAvL6B,SAuLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxLhB,OAoLO,W;O;KAnMX,C;uFAYA,yB;MAAA,+D;MA0LA,gD;MAiLA,uC;QASW,kBAAU,gB;QAwLD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAZL6B,SAyLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1LhB,OA4LO,W;O;KArMX,C;qGAYA,yB;MAAA,+D;MA4DA,gD;MA5DA,uC;QAYW,kBAAiB,gB;QA2DR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA5DoC,SA4DzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA7DhB,OA+DO,W;O;KA3EX,C;qGAeA,yB;MAAA,+D;MA+DA,gD;MA/DA,uC;QAYW,kBAAiB,gB;QA8DR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA/DoC,SA+DzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhEhB,OAkEO,W;O;KA9EX,C;qGAeA,yB;MAAA,+D;MAkEA,gD;MAiEA,uC;QAYW,kBAAiB,gB;QAIER,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAIeOC,SAkEzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAnEhB,OAqEO,W;O;KAjFX,C;qGAeA,yB;MAAA,+D;MAqEA,gD;MArEA,uC;QAYW,kBAAiB,gB;QAoER,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WArEOC,SAqEzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtEhB,OAweO,W;O;KApFX,C;yGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACSM;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACSM;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACSM;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACSM;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FACa,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FACa,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;uFACa,yB;MAAA,wE;MA4HA,+D;MA5HA,yC;QAYW,kBAAU,oB;QA4HD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA7HoD,WA6H1C,CAAY,OAAZ,C;U/B59IP,U;UADP,Y+B89Ie,W/B99IH,W+B89IwB,G/B99IxB,C;UACL,IAAI,aAAJ,C;YACH,a+B49IuC,gB;YAA5B,W/B39IX,a+B29IgC,G/B39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+Bw9IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA/HT,OAIIO,W;O;KA7IX,C;uFAeA,yB;MAAA,wE;MAiIA,+D;MAjIA,yC;QAYW,kBAAU,oB;QAIID,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAlIqD,WakI3C,CAAY,OAAZ,C;U/Bh/IP,U;UADP,Y+Bk/Ie,W/BI/IH,W+Bk/IwB,G/BI/IxB,C;UACL,IAAI,aAAJ,C;YACH,a+Bg/IuC,gB;YAA5B,W/B/+IX,a+B++IgC,G/B/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B4+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApIT,OAsIO,W;O;KAIJX,C;sFAeA,yB;MAAA,wE;MAsIA,+D;MatIA,y

C;QAYW,kBAAU,oB;QAsID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAvIqD,WAuI3C,CAAY,OAAZ,C;U/BpgJP,U;UADP,Y+BsgJe,W/BtgJH,W+BsgJwB,G/BtgJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BogJuC,gB;YAA5B,W/BngJX,a+BmgJgC,G/BngJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BggJA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAZIT,OA2IO,W;O;KAvJX,C;uFAeA,yB;MAAA,wE;MA2IA,+D;MA3IA,yC;QAYW,kBAAU,oB;QA2ID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA5IsD,WA4I5C,CAAY,OAAZ,C;U/BxhJP,U;UADP,Y+B0hJe,W/B1hJH,W+B0hJwB,G/B1hJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BwhJuC,gB;YAA5B,W/BvhJX,a+BuhJgC,G/BvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BohJA,iB;UACA,IAAK,WAAL,OAAJ,C;;QA9IT,OAgJO,W;O;KA5JX,C;uFAeA,yB;MAAA,wE;MAgJA,+D;MAhJA,yD;QAaW,kBAAU,oB;QA9JD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAJjID,WaiJvC,CAAY,OAAZ,C;U/B7iJP,U;UADP,Y+B+iJe,W/B/iJH,W+B+iJwB,G/B/iJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B6iJuC,gB;YAA5B,W/B5iJX,a+B4iJgC,G/B5iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+ByiJA,iB;UACA,IAAK,WAnJyD,cAmJrD,CAAe,OAAf,CAAJ,C;;QAnJT,OAgJO,W;O;KAIKX,C;uFAGBA,yB;MAAA,wE;MAqJA,+D;MArJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAtjID,WAsJvC,CAAY,OAAZ,C;U/BlkJP,U;UADP,Y+BokJe,W/BpkJH,W+BokJwB,G/BpkJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BkkJuC,gB;YAA5B,W/BjkJX,a+BikJgC,G/BjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B8jJA,iB;UACA,IAAK,WaxJyD,cAwJrD,CAAe,OAAf,CAAJ,C;;QAxJT,OA0JO,W;O;KAvKX,C;uFAGBA,yB;MAAA,wE;MA0JA,+D;MA1JA,yD;QAaW,kBAAU,oB;QA0JD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA3JiD,WA2JvC,CAAY,OAAZ,C;U/BvIJP,U;UADP,Y+BvIJe,W/BzIjH,W+BvIjwB,G/BzIjxB,C;UACL,IAAI,aAAJ,C;YACH,a+BulJuC,gB;YAA5B,W/BtlJX,a+BslJgC,G/BtlJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BmlJA,iB;UACA,IAAK,WA7JyD,cA6JrD,CAAe,OAAf,CAAJ,C;;QA7JT,OA+JO,W;O;KA5KX,C;uFAGBA,yB;MAAA,wE;MA+JA,+D;MA/JA,yD;QAaW,kBAAU,oB;QA+JD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAhKiD,WAgKvC,CAAY,OAAZ,C;U/B5mJP,U;UADP,Y+B8mJe,W/B9mJH,W+B8mJwB,G/B9mJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B4mJuC,gB;YAA5B,W/B3mJX,a+B2mJgC,G/B3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BwmJA,iB;UACA,IAAK,WAlKyD,cAkKrD,CAAe,OAAf,CAAJ,C;;QAiKT,OAoKO,W;O;KAjLX,C;2FAGBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/B59IP,U;UADP,Y+B89Ie,W/B99IH,W+B89IwB,G/B99IxB,C;UACL,IAAI,aAAJ,C;YACH,a+B49IuC,gB;YAA5B,W/B39IX,a+B29IgC,G/B39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+Bw9IA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/Bh/IP,U;UADP,Y+Bk/Ie,W/Bi/IH,W+Bk/IwB,G/Bi/IxB,C;UACL,IAAI,aAAJ,C;YACH,a+Bg/IuC,gB;YAA5B,W/B+IX,a+B+IgC,G/B+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B4+IA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BpgJP,U;UADP,Y+BsgJe,W/BtgJH,W+BsgJwB,G/BtgJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BogJuC,gB;YAA5B,W/BngJX,a+BmgJgC,G/BngJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BggJA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BxhJP,U;UADP,Y+B0hJe,W/B1hJH,W+B0hJwB,G/B1hJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BwhJuC,gB;YAA5B,W/BvhJX,a+BuhJgC,G/BvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BohJA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/B7iJP,U;UADP,Y+B+iJe,W/B/iJH,W+B+iJwB,G/B/iJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B6iJuC,gB;YAA5B,W/B5iJX,a+B4iJgC,G/B5iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+ByiJA,iB;UACA,IAAK,WAAL,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BlkJP,U;UADP,Y+BokJe,W/BpkJH,W+BokJwB,G/BpkJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BkkJuC,gB;YAA5B,W/BjkJX,a+BikJgC,G/BjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B8jJA,iB;UACA,IAAK,WAAL,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BvIJP,U;UADP,Y+BvIJe,W/BzIjH,W+BvIjwB,G/BzIjxB,C;UACL,IAAI,aAAJ,C;YACH,a+BulJuC,gB;YAA5B,W/BtlJX,a+BslJgC,G/BtlJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BmlJA,iB;UACA,IAAK,WAAL,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAA

A,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAGb,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAAY,OAAZ,C;U/B5mJP,U;UADP,Y+B8mJe,W/B9mJH,W+B8mJwB,G/B9mJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B4mJuC,gB;YA A5B,W/B3mJX,a+B2mJgC,G/B3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BwmJA,iB;UACA,IAAK,WAAI,e AAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAlBX,C;+EAqBA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,c AAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAvKiB,SAuKb,CAAU,IAAV,CA AJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAs KA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAv KhB,OAwKO,W;O;KAILX,C;8EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAA A,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwK O,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb ,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAI LX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAaA,eAAa,cAAb,C;QAqDP,gB;QADb,YAAAY,C;QACC,2B;Q AAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAtDwB,SAsDpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB ,CAAJ,C;;QAtDhB,OAuDO,W;O;KAjEX,C;6FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAaA,eAAa,cAAb,C;QA wDP,gB;QADb,YAAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WAZDwB,SAYDpB,EAAU, cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAZDhB,OA0DO,W;O;KApEX,C;6FAaA,yB;MAAA,gE;MAA A,uC;QAUW,kBAaA,eAAa,cAAb,C;QA2DP,gB;QADb,YAAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA CT,WAAAY,WA5DwB,SA4DpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA5DhB,OA6DO,W;O ;KAvEX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAaA,eAAa,cAAb,C;QA8DP,gB;QADb,YAAAY,C;QACC, 2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WA/DwB,SA+DpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,I AAnB,CAAJ,C;;QA/DhB,OAgEO,W;O;KA1EX,C;iGAaA,6C;MAWiB,UACiB,M;MAF9B,YAAAY,C;MACC,2B; MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C; ;MACHB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M;MAF9B,YAAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAA a,sB;QACT,WAAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;iGAG X,6C;MAWiB,UACiB,M;MAF9B,YAAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI, WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M; MAF9B,YAAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI,WAAU,cAAV,EAAU,sBA AV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;Q AAa,sB;QACT,WAAAY,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B; MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;mFAGX,6 C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI,UAAU,IAAV,CAAJ,C;;MACHB ,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAAY,WAAI,UAAU,I AAV,CAAJ,C;;MACHB,OAAO,W;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8B AAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MA AA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IA P9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGb,cAAhB,C;QA AgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;+EAGX,gC;MASo B,Q;MAAA,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,O AAO,K;;MACtD,OAAO,I;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QAAM,IA AI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;+EAGX,gC;MASoB,Q;MAAA,2B;M AAhB,OAAGb,cAAhB,C;QAAGb,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD, OAAO,I;K;+EAGX,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C ;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iB AAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,gC; MASoB,Q;MAAA,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,O AAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QAAM,IA AI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB, OAAGb,cAAhB,C;QAAGb,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+E





AAO,sBAAP,WAAgB,IAAhB,C;;K;IAGvB,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;mFAGX,yB;MA9gEI,8D;MA8gEJ,sC;QAMW,sB;;UAuCP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA7jEgB,cAAR,iBAAQ,C;UA8jEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3CmB,QA2CJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ce,QA8CP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApDP,yB;O;KANJ,C;mFASA,yB;MA/gEI,8D;MA+gEJ,sC;QAMW,sB;;UAuDP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA9kEgB,cAAR,iBAAQ,C;UA+kEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3DmB,QA2DJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9De,QA8DP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApEP,yB;O;KANJ,C;mFASA,yB;MAhhEI,8D;MAghEJ,sC;QAMW,sB;;UAuEP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA/IEgB,cAAR,iBAAQ,C;UAGmEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3EmB,QA2EJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ee,QA8EP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApFP,yB;O;KANJ,C;mFASA,yB;MAjhEI,8D;MAihEJ,sC;QAMW,sB;;UAuFP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBAhnEgB,cAAR,iBAAQ,C;UAinEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3FmB,QA2FJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Fe,QA8FP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApGP,yB;O;KANJ,C;+FASA,yB;MALjEI,8D;MAkjEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA7jEgB,cA6jEA,SA7jER,QAAQ,C;QA8jEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAnkEI,8D;MAmkEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA9kEgB,cA8kEA,SA9kER,QAAQ,C;QA+kEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAplEI,8D;MAolEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA/IEgB,cA+IEA,SA/IER,QAAQ,C;QAGmEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MArmEI,8D;MAqmEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAhnEgB,cAgnEA,SAhnER,QAAQ,C;QAinEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MAtpEI,8D;MpBnwHJ,iB;MoBy5LA,sC;QAGBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtqEG,cAAR,iBAAQ,C;QAsqEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn6LG,MAAO,KoBm6LO,QpBn6LP,EoBm6LiB,CpBn6LjB,C;;QoBq6Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MARqEI,8D;MpB3wHJ,iB;MoBg7LA,sC;QAGBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArrEG,cAAR,iBAAQ,C;QAqrEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB17LG,MAAO,KoB07LO,QpB17LP,EoB07LiB,CpB17LjB,C;;QoB47Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAprEI,8D;MpBnxHJ,iB;MoBu8LA,sC;QAGBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAPsEG,cAAR,iBAAQ,C;QAosEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBj9LG,MAAO,KoBi9LO,QpBj9LP,EoBi9LiB,CpBj9LjB,C;;QoBm9Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAnsEI,8D;MpB3xHJ,iB;MoB89LA,sC;QAGBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAntEG,cAAR,iBAAQ,C;QAmtEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx+LG,MAAO,KoBw+L

O,QpBx+LP,EoBw+LiB,CpBx+LjB,C;;QoB0+Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAIvEI,8D;M  
pB9wHJ,iB;MoBggMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,C  
AAL,CAAT,C;QACF,OAlwEG,cAAR,iBAAQ,C;QakwEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAA  
L,CAAT,C;UACR,WpB1gMG,MAAO,KoB0gMO,QpB1gMP,EoB0gMiB,CpB1gMjB,C;;QoB4gMd,OAAO,Q;O;  
KApBX,C;mFAuBA,yB;MAAA,sE;MAjwEI,8D;MpBtxHJ,iB;MoBuhMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;  
UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAJxEG,cAAR,iBAAQ,C;QAixEhB,aA  
AU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBjiMG,MAAO,KoBiiMO,QpBjiMP,EoBii  
MiB,CpBjiMjB,C;;QoBmiMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhxEI,8D;MpB9xHJ,iB;MoB8i  
MA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC  
F,OAhYEG,cAAR,iBAAQ,C;QAgYehB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,W  
pBxjMG,MAAO,KoBwjMO,QpBxjMP,EoBwjMiB,CpBxjMjB,C;;QoB0jMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;  
MAAA,sE;MA/xEI,8D;MpBtyHJ,iB;MoBqkMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB  
,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/yEG,cAAR,iBAAQ,C;QA+yEhB,aAAU,CAAV,iB;UACI,QAA  
Q,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/kMG,MAAO,KoB+kMO,QpB/kMP,EoB+kMiB,CpB/kMjB,C;;Qo  
BilMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA90EI,8D;MA80EJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C  
;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA51EG,cAAR,iBAAQ,C;QA41EhB,a  
AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA  
AW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA71EI,8D;MA61EJ,sC;QAcIB,Q;QAFb,IAAI,  
mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA32EG,cAAR,iBAAQ,C;Q  
A22EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y  
ACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA52EI,8D;MA42EJ,sC;QAcIB,Q;QA  
Fb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA13EG,cAAR,iB  
AAQ,C;QA03EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K  
AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA33EI,8D;MA23EJ,sC;QAc  
iB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz4EG,cA  
AR,iBAAQ,C;QAY4EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C  
AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA16EI,8D;MpBnwHJ,iB;MoB6q  
MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,O  
Ax7EG,cAAR,iBAAQ,C;QAw7EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBr  
rMG,MAAO,KoBqrMO,QpBrrMP,EoBqrMiB,CpBrrMjB,C;;QoBurMd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAv7  
EI,8D;MpB3wHJ,iB;MoBksMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAA  
K,CAAL,CAAT,C;QACF,OA8r8EG,cAAR,iBAAQ,C;QAq8EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,C  
AAL,CAAT,C;UACR,WpB1sMG,MAAO,KoB0sMO,QpB1sMP,EoB0sMiB,CpB1sMjB,C;;QoB4sMd,OAAO,Q;O  
;KAlBX,C;+FAqBA,yB;MAp8EI,8D;MpBnxHJ,iB;MoButMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,  
I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA19EG,cAAR,iBAAQ,C;QAk9EhB,aAAU,CAAV,iB;UA  
CI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/tMG,MAAO,KoB+tMO,QpB/tMP,EoB+tMiB,CpB/tMjB,  
C;;QoBiuMd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAj9EI,8D;MpB3xHJ,iB;MoB4uMA,sC;QAcIB,Q;QAFb,IAAI,  
mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/9EG,cAAR,iBAAQ,C;QA+9  
EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBpvMG,MAAO,KoBovMO,QpBp  
vMP,EoBovMiB,CpBpvMjB,C;;QoBsvMd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA9/EI,8D;MpB9wHJ,iB;MoB4w  
MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,O  
A5gFG,cAAR,iBAAQ,C;QA4gFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp  
xMG,MAAO,KoBoxMO,QpBpxMP,EoBoxMiB,CpBpxMjB,C;;QoBsxMd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;M  
A3gFI,8D;MpBtxHJ,iB;MoBiyMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBA  
AK,CAAL,CAAT,C;QACF,OAzhFG,cAAR,iBAAQ,C;QAYhFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,  
CAAL,CAAT,C;UACR,WpBzyMG,MAAO,KoByyMO,QpBzyMP,EoByyMiB,CpBzyMjB,C;;QoB2yMd,OAAO,Q  
;O;KAlBX,C;+FAqBA,yB;MAxhFI,8D;MpB9xHJ,iB;MoBszMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAA  
O,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtiFG,cAAR,iBAAQ,C;QAsiFhB,aAAU,CAAV,iB;U

ACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9zMG,MAAO,KoB8zMO,QpB9zMP,EoB8zMiB,CpB9z MjB,C;;QoBg0Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MArIFl,8D;MpBtyHJ,iB;MoB20MA,sC;QAcIB,Q;QAFb,I AAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9jFG,cAAR,iBAAQ,C;Q AmjFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn1MG,MAAO,KoBm1MO, QpBn1MP,EoBm1MiB,CpBn1MjB,C;;QoBq1Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAlIFI,8D;MAkIFJ,sC;QA YiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IFG,cAA R,iBAAQ,C;QA8IFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CA AX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAIFI,8D;MA+IFJ,sC;QAYiB,Q;Q AFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mFG,cAAR,iBA AQ,C;QA2mFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA5mFI,8D;MA4mFJ,sC;QAYiB,Q;QAF b,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxnFG,cAAR,iBAAQ, C;QAwnFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAznFI,8D;MAynFJ,sC;QAYiB,Q;QAFb,IAA I,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAroFG,cAAR,iBAAQ,C;QAq oFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YAC I,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MAtqFI,8D;MASqFJ,kD;QAcIB,Q;QAFb,IA AI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAprFG,cAAR,iBAAQ,C; QAorFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EA AkB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,s E;MArrFI,8D;MAqrFJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAA L,CAAT,C;QACF,OA9sFG,cAAR,iBAAQ,C;QAmsFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C AAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGn B,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MApsFI,8D;MAosFJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAA e,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAItFG,cAAR,iBAAQ,C;QaktFhB,aAAU,CA AV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX, GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAntFI,8D;MAmt FJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF, OAjuFG,cAAR,iBAAQ,C;QAiuFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI, UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApB X,C;uGAuBA,yB;MAlwFI,8D;MAkwFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAA S,sBAAK,CAAL,CAAT,C;QACF,OA9wFG,cAAR,iBAAQ,C;QA8wFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,s BAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MA/wFI,8D;MA+wFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ, C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3xFG,cAAR,iBAAQ,C;QA2xFhB,aA AU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB, CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MA5xFI,8D;MA4xFJ, kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxy FG,cAAR,iBAAQ,C;QAwyFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C; uGAqBA,yB;MAzyFI,8D;MAyyFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sB AAK,CAAL,CAAT,C;QACF,OA9zFG,cAAR,iBAAQ,C;QAqzFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK, CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C; ;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MActB,UAAU,sB AAK,CAAL,C;MACG,OA9IFG,gBAAR,iBAAQ,C;MA8IFhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C; QACR,InC5mN8D,YmC4mN1D,GnC5mN2E,KAAjB,EmC4mNpD,CnC5mNiF,KAA7B,CmC4mN1D,IAAJ,C;UA Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MActB,UAAU,sB AAK,CAAL,C;MACG,OA92FG,gBAAR,iBAAQ,C;MAq2FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C

;QACR,InBnnN+D,amBmnN3D,GnBnnN6E,KAAiB,EmBmnNrD,CnBnnNmF,KAA9B,CmBmnN3D,IAAJ,C;UA  
Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,s  
BAAK,CAAL,C;MACG,OA52FG,gBAAR,iBAAQ,C;MA42FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C  
;QACR,IpC1pN4E,0BoC0pNxE,GpC/6M8B,KAAAL,GAAiB,GA3O8B,EoC0pNIE,CpC/6MwB,KAAAL,GAAiB,GA  
3O8B,CoC0pNxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;Q  
AAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OA3n3FG,gBAAR,iBAAQ,C;MAm3FhB,aAAU,CAAV,iB  
;QACI,QAAQ,sBAAK,CAAL,C;QACR,IICjqN6E,0BkCiqNzE,GIC77M8B,KAAAL,GAAiB,KApO+B,EkCiqNnE,C  
IC77MwB,KAAAL,GAAiB,KApO+B,CkCiqNzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAKI,  
OAAO,4BAAc,UAAc,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAc,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAc,  
C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAc,C;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;M  
ACTb,UAAU,sBAAK,CAAL,C;MACG,OA17FG,gBAAR,iBAAQ,C;MA07FhB,aAAU,CAAV,iB;QACI,QAAQ,s  
BAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;  
MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,  
C;MACG,OAj8FG,gBAAR,iBAAQ,C;MAi8FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,U  
AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;  
MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAx8FG,gBAAR,iB  
AAQ,C;MAw8FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,C  
AAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mB  
AAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OA/8FG,gBAAR,iBAAQ,C;MA+8FhB,aAAU,CA  
AV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,  
C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2  
B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;mFAGX,yB;MA9gGI,8D;MA8gGJ,sC;QAMW,sB;;UAuCP,  
IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA7jGgB,cAAR,iBAAQ,C;U  
A8jGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3CmB,QA2CJ,CAAS,OAAT,C;UACf,a  
AAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ce,QA8CP,CAAS,CAAT,C;YACR,IA  
AI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApDP,yB;O;KANJ,C;mFA  
SA,yB;MA/gGI,8D;MA+gGJ,sC;QAMW,sB;;UAuDP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,s  
BAAK,CAAL,C;UACd,gBA9kGgB,cAAR,iBAAQ,C;UA+kGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP  
,uB;;UACpB,eA3DmB,QA2DJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAA  
L,C;YACR,QA9De,QA8DP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAA  
W,C;;UAGnB,qBAAO,O;;QApEP,yB;O;KANJ,C;mFASA,yB;MAhhGI,8D;MAghGJ,sC;QAMW,sB;;UAuEP,IA  
AI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA/IGgB,cAAR,iBAAQ,C;UAg  
mGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3EmB,QA2EJ,CAAS,OAAT,C;UACf,aA  
AU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ee,QA8EP,CAAS,CAAT,C;YACR,IAAI,  
2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApFP,yB;O;KANJ,C;mFASA  
,yB;MAjhGI,8D;MAihGJ,sC;QAMW,sB;;UAuFP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAA  
K,CAAL,C;UACd,gBAhnGgB,cAAR,iBAAQ,C;UAinGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;  
UACpB,eA3FmB,QA2FJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;  
YACR,QA9Fe,QA8FP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;  
UAGnB,qBAAO,O;;QApGP,yB;O;KANJ,C;+FASA,yB;MALjGI,8D;MAkjGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,O  
AAO,I;QACTb,cAAc,sBAAK,CAAL,C;QACd,gBA7jGgB,cA6jGA,SA7jGR,QAAQ,C;QA8jGhB,IAAI,cAAa,CA  
AjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAA  
AK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA  
W,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAnkGI,8D;MAmkGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAA  
O,I;QACTb,cAAc,sBAAK,CAAL,C;QACd,gBA9kGgB,cA8kGA,SA9kGR,QAAQ,C;QA+kGhB,IAAI,cAAa,CAAj  
B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAA  
K,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA  
W,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAplGI,8D;MAolGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO

,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA/IGgB,cA+IGA,SA/IGR,QAAQ,C;QAgmGhB,IAAI,cAAa,CAAjB,C  
;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,C  
AAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;  
;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MArmGI,8D;MAqmGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;  
QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAhnGgB,cAgnGA,SAhnGR,QAAQ,C;QAinGhB,IAAI,cAAa,CAAjB,C;  
UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CA  
AL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;  
QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MAtpGI,8D;MpB/iHJ,iB;MoBqsNA,sC;QAgBiB,Q;QAFb  
,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtqGG,cAAR,iBAAQ  
,C;QAsqGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/sNG,MAAO,KoB+sN  
O,QpB/sNP,EoB+sNiB,CpB/sNjB,C;;QoBitNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MArqGI,8D;MpB  
vjHJ,iB;MoB4tNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,  
CAAT,C;QACF,OArGG,cAAR,iBAAQ,C;QAqrGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAA  
T,C;UACR,WpBtuNG,MAAO,KoBsuNO,QpBtuNP,EoBsuNiB,CpBtuNjB,C;;QoBwuNd,OAAO,Q;O;KApBX,C;m  
FAuBA,yB;MAAA,sE;MAprGI,8D;MpB/jHJ,iB;MoBmvNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM  
,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApGG,cAAR,iBAAQ,C;QAosGhB,aAAU,CAAV,iB;  
UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB7vNG,MAAO,KoB6vNO,QpB7vNP,EoB6vNiB,CpB7v  
NjB,C;;QoB+vNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAnsGI,8D;MpBvkHJ,iB;MoB0wNA,sC;QAg  
BiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAntGG,c  
AAR,iBAAQ,C;QAmtGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBpxNG,M  
AAO,KoBoxNO,QpBpxNP,EoBoxNiB,CpBpxNjB,C;;QoBsxNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;  
MAlvGI,8D;MpB1jHJ,iB;MoB4yNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SA  
AS,sBAAK,CAAL,CAAT,C;QACF,OAlwGG,cAAR,iBAAQ,C;QAkwGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,  
sBAAK,CAAL,CAAT,C;UACR,WpBtzNG,MAAO,KoBszNO,QpBtzNP,EoBszNiB,CpBtzNjB,C;;QoBwzNd,OAA  
O,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjwGI,8D;MpBlkHJ,iB;MoBm0NA,sC;QAgBiB,Q;QAFb,IAAI,mB  
AAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAJGG,cAAR,iBAAQ,C;QAix  
GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB70NG,MAAO,KoB60NO,QpB7  
0NP,EoB60NiB,CpB70NjB,C;;QoB+0Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhxGI,8D;MpB1kHJ  
,iB;MoB01NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CA  
AT,C;QACF,OAhGG,cAAR,iBAAQ,C;QAgyGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,  
C;UACR,WpBp2NG,MAAO,KoBo2NO,QpBp2NP,EoBo2NiB,CpBp2NjB,C;;QoBs2Nd,OAAO,Q;O;KApBX,C;m  
FAuBA,yB;MAAA,sE;MA/xGI,8D;MpBlIHJ,iB;MoBi3NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,  
6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/yGG,cAAR,iBAAQ,C;QA+yGhB,aAAU,CAAV,iB;  
UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB33NG,MAAO,KoB23NO,QpB33NP,EoB23NiB,CpB33  
NjB,C;;QoB63Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA90GI,8D;MA80GJ,sC;QAcIB,Q;QAFb,IAA  
I,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA51GG,cAAR,iBAAQ,C;  
QA41GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;  
YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA71GI,8D;MA61GJ,sC;QAcIB,Q;Q  
AFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA32GG,cAAR,iB  
AAQ,C;QA22GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,  
KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA52GI,8D;MA42GJ,sC;Q  
AcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA13GG  
,cAAR,iBAAQ,C;QA03GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAA  
W,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA33GI,8D;MA  
23GJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC  
F,OAz4GG,cAAR,iBAAQ,C;QAy4GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,I  
AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA16GI,8D;MpB  
/iHJ,iB;MoBy9NA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAA

T,C;QACF,OAx7GG,cAAR,iBAAQ,C;QA w7GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBj+NG,MAAO,KoBi+NO,QpBj+NP,EoBi+NiB,CpBj+NjB,C;;QoBm+Nd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAv7GI,8D;MpBvjHJ,iB;MoB8+NA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAr8GG,cAAR,iBAAQ,C;QAq8GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBt/NG,MAAO,KoBs/NO,QpBt/NP,EoBs/NiB,CpBt/NjB,C;;QoBw/Nd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAp8GI,8D;MpB/jHJ,iB;MoBmgOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAI9GG,cAAR,iBAAQ,C;QAk9GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB3gOG,MAAO,KoB2gOO,QpB3gOP,EoB2gOiB,CpB3gOjB,C;;QoB6gOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAj9GI,8D;MpBvkHJ,iB;MoBwhOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9GG,cAAR,iBAAQ,C;QA+9GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBhiOG,MAAO,KoBgiOO,QpBhiOP,EoBgiOiB,CpBhiOjB,C;;QoBkiOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA9/GI,8D;MpB1jHJ,iB;MoBwjOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5gHG,cAAR,iBAAQ,C;QA4gHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBhkOG,MAAO,KoBgkOO,QpBhkOP,EoBgkOiB,CpBhkOjB,C;;QoBkkOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3gHI,8D;MpBlkHJ,iB;MoB6kOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzhHG,cAAR,iBAAQ,C;QAyhHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrlOG,MAAO,KoBqlOO,QpBrlOP,EoBqlOiB,CpBrlOjB,C;;QoBulOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAxhHI,8D;MpB1kHJ,iB;MoBkmOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtiHG,cAAR,iBAAQ,C;QAsiHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1mOG,MAAO,KoB0mOO,QpB1mOP,EoB0mOiB,CpB1mOjB,C;;QoB4mOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAriHI,8D;MpBlIHJ,iB;MoBunOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnjHG,cAAR,iBAAQ,C;QAmjHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/nOG,MAAO,KoB+nOO,QpB/nOP,EoB+nOiB,CpB/nOjB,C;;QoBioOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAIHI,8D;MAkiHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IHG,cAAR,iBAAQ,C;QA8IHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA/IHI,8D;MA+IHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mHG,cAAR,iBAAQ,C;QA2mHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA5mHI,8D;MA4mHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxnHG,cAAR,iBAAQ,C;QAwnHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAznHI,8D;MAynHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArHG,cAAR,iBAAQ,C;QAqoHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MAAtqHI,8D;MASqHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAprHG,cAAR,iBAAQ,C;QAorHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAlB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MArrHI,8D;MAqrHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnsHG,cAAR,iBAAQ,C;QAmsHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAlB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MApsHI,8D;MAosHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAltHG,cAAR,iBAAQ,C;QAktHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAlB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAntHI,8D;MAmtHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;

QACF,OAjuHG,cAAR,iBAAQ,C;QAiuHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MAIwHI,8D;MAkwHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9wHG,cAAR,iBAAQ,C;QA8wHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MA/wHI,8D;MA+wHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3xHG,cAAR,iBAAQ,C;QA2xHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MA5xHI,8D;MA4xHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxyHG,cAAR,iBAAQ,C;QAwyHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAzyHI,8D;MAyyHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArzHG,cAAR,iBAAQ,C;QAqzHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OA91HG,gBAAR,iBAAQ,C;MA81HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InC5mP8D,YmC4mP1D,GnC5mP2E,KAAjB,EmC4mPpD,CnC5mPiF,KAA7B,CmC4mP1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OAr2HG,gBAAR,iBAAQ,C;MAq2HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InBnnP+D,amBmnP3D,GnBnnP6E,KAAIB,EmBmnPrD,CnBnnPmF,KAA9B,CmBmnP3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OA52HG,gBAAR,iBAAQ,C;MA42HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpC1pP4E,0BoC0pPxE,GpC/6O8B,KAAL,GAAiB,GA3O8B,EoC0pPIE,CpC/6OwB,KAAL,GA AiB,GA3O8B,CoC0pPxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OAn3HG,gBAAR,iBAAQ,C;MAm3HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IICjqP6E,0BkCiqPzE,GIC77O8B,KAAL,GAAiB,KApO+B,EkCiqPnE,CIC77OwB,KAAL,GAAiB,KApO+B,CkCiqPzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OA17HG,gBAAR,iBAAQ,C;MA07HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OAj8HG,gBAAR,iBAAQ,C;MAi8HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OAx8HG,gBAAR,iBAAQ,C;MAw8HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;qFAGX,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K



;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB  
;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,  
gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAAr  
C,gB;K;mGAGJ,6B;MAtREiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,c  
AAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAgsEnB,gB;K;mGAGJ,6B;MAtREiB,gB;MADb,YAAY,C;MACC,2B;  
MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAgsEnB,gB;K;mGA  
GJ,6B;MAtREiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,s  
BAAP,WAAgB,IAAhB,C;;MAgsEnB,gB;K;mGAGJ,6B;MAtREiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,  
cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAgsEnB,gB;K;qFAGJ,yB;MAAA,  
4F;MA9qII,8D;MA8qIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kB  
AAkB,sBAAK,CAAL,C;QACD,OAJsID,cAAR,iBAAQ,C;QAisIhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV  
,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MA+rII,8D;MA+rII,u  
C;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;  
QACD,OAltID,cAAR,iBAAQ,C;QaktIhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,  
CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAhtII,8D;MAgtII,uC;QAmBqB,Q;QAHjB,I  
AAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAnuID,cAAR,i  
BAAQ,C;QAmulhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,O  
AAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAjuII,8D;MAiuIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,  
MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OApvID,cAAR,iBAAQ,C;QAovIhB,iB  
AAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;m  
GAyBA,yB;MAAA,4F;MALxII,8D;MAkxIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+B  
AA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OArYID,cAAR,iBAAQ,C;QAqyIhB,iBAAc,CAAd,yB;UACI,  
cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAyB  
A,yB;MAAA,4F;MAnyII,8D;MAmyIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9  
B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAtzID,cAAR,iBAAQ,C;QAszIhB,iBAAc,CAAd,yB;UACI,cAAc,  
UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAyBA,yB;  
MAAA,4F;MApzII,8D;MAozIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;Q  
ACV,kBAAkB,sBAAK,CAAL,C;QACD,OAv0ID,cAAR,iBAAQ,C;Qau0IhB,iBAAc,CAAd,yB;UACI,cAAc,UAA  
U,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAyBA,yB;MAA  
A,4F;MAr0II,8D;MAq0IJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,k  
BAAkB,sBAAK,CAAL,C;QACD,OAx1ID,cAAR,iBAAQ,C;QAw1IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KA  
AV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAyBA,yB;Mat3II,8D;  
MA53IJ,uC;QakBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAx4  
ID,cAAR,iBAAQ,C;Qaw4IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,K  
AAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAwBA,yB;Mat4II,8D;MA54IJ,uC;QakBqB,Q;QAHjB,IAAI,  
mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAx5ID,cAAR,iBAAQ,C;Qaw5IhB,iBAAc  
,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;  
KArBX,C;+GAwBA,yB;Mat5II,8D;MA55IJ,uC;QakBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBA  
AkB,sBAAK,CAAL,C;QACD,OAx6ID,cAAR,iBAAQ,C;Qaw6IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV  
,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAwBA,yB;Mat6II,8D;M  
As6IJ,uC;QakBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAx7ID  
,cAAR,iBAAQ,C;Qaw7IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAA  
L,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;iGAwBA,yB;Mat9II,8D;MA59IJ,uC;QAmBqB,Q;QAHjB,IAAI,mB  
AAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAz+ID,cAAR,iBAAQ,C;QAY+IhB,iBAAc,CA  
Ad,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAyBA,y  
B;MAv+II,8D;MAu+IJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,  
C;QACD,OAl/ID,cAAR,iBAAQ,C;QA0/IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KA  
AL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAyBA,yB;MAx/II,8D;MAw/IJ,uC;QAmBqB,Q;QAHjB,IAAI,m

BAAJ,C;UACI,OAAO,I;QACX,kBAaKb,sBAaK,CAAL,C;QACD,OA3gJD,cAAR,iBAAQ,C;QA2gJhB,iBAAc,C  
AAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAaK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA  
,yB;MAzgJI,8D;MAygJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAaKb,sBAaK,CAA  
L,C;QACD,OA5hJD,cAAR,iBAAQ,C;QA4hJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAaK,  
KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA1jJI,8D;MA0jJJ,uC;QakB0B,UAE  
U,M;QAJhC,YA1kJgB,cAAR,iBAAQ,C;QA2kJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;  
QACrB,kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cA  
AJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA3kJI,8D  
;MA2kJJ,uC;QakB0B,UAEU,M;QAJhC,YA3IJgB,cAAR,iBAAQ,C;QA4IJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAA  
M,mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UAC  
I,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,  
yB;MAAA,4F;MA5IJI,8D;MA4IJJ,uC;QakB0B,UAEU,M;QAJhC,YA5mJgB,cAAR,iBAAQ,C;QA6mJhB,IAAI,Q  
AAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,O  
AAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OA  
AO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA7mJI,8D;MA6mJJ,uC;QakB0B,UAEU,M;QAJhC,YA7nJgB,cAA  
R,iBAAQ,C;QA8nJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YA  
AJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EA  
AwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;6GAyBA,yB;MAAA,4F;MA9pJI,8D;MA8pJJ,uC;QakB0B,Q;Q  
AFtB,YA9qJgB,cAAR,iBAAQ,C;QA+qJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,  
kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sB  
AAI,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAhrJ  
I,8D;MAgrJJ,uC;QakB0B,Q;QAFtB,YAhsJgB,cAAR,iBAAQ,C;QAisJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,  
mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,c  
AAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6  
GA0BA,yB;MAAA,4F;MAIsJI,8D;MAksJJ,uC;QakB0B,Q;QAFtB,YAltJgB,cAAR,iBAAQ,C;QAmTJhB,IAAI,QA  
AQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OA  
AO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;Q  
AEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAptJI,8D;MAotJJ,uC;QakB0B,Q;QAFtB,YApuJgB,cAAR  
,iBAAQ,C;QAquJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKb,uBAAI,YAA  
J,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EA  
A6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;yHA0BA,yB;MATwJI,8D;MASwJJ,uC;QaiB0B,Q;QAF  
tB,YArxJgB,cAAR,iBAAQ,C;QAsxJhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,E  
AAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6  
B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,yB;MAxyJI,8D;MAwyJJ,uC;QaiB0B,Q;QAFtB,  
YAtyJgB,cAAR,iBAAQ,C;QAuyJhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,EA  
AI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,  
WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,yB;MAxyJI,8D;MAwyJJ,uC;QaiB0B,Q;QAFtB,Y  
AvzJgB,cAAR,iBAAQ,C;QAwzJhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,EAAI  
,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,W  
AA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,yB;MAzzJI,8D;MAyzJJ,uC;QaiB0B,Q;QAFtB,YAx0  
JgB,cAAR,iBAAQ,C;QAY0JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,EAAI,oB  
AAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,WAA  
7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA12JI,8D;MA02JJ,uC;QakB0B,UAEU,M;QAJhC,  
YA13JgB,cAAR,iBAAQ,C;QA23JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,EA  
AI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,  
WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA33JI,8D;MA23JJ,uC;QakB0B,UAEU,M;QAJhC,YA  
34JgB,cAAR,iBAAQ,C;QA44JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKb,uBAAI,YAAJ,EAAI,o  
BAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAA

xB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA54JI,8D;MA44JJ,uC;QakB0B,UAEU,M;QAJhC,YA55JgB,cAAR,iBAAQ,C;QA65JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA75JI,8D;MA65JJ,uC;QakB0B,UAEU,M;QAJhC,YA76JgB,cAAR,iBAAQ,C;QA86JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBrRO,W;QqBstRP,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB9uRO,W;QqB+uRP,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBvwRO,W;QqBwwRP,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QA A5C,arBhyRO,W;QqBiyRP,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;6GAyBA,yB;MAAA,gD;MAAA,gE;MAIIKI,0D;MAkiKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QAC c,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB1zRO,W;QqB2zRP,kBAakB,O;QACJ,OArmK E,YAAR,iBAAQ,C;QAqmKF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB, EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAvBX,C;6GA0BA,yB; MAAA,gD;MAAA,gE;MApmKI,0D;MAomKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OA AP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBp1RO,W;QqBq1RP,kBAakB,O;Q ACJ,OAvnKE,YAAR,iBAAQ,C;QAunKF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAi B,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAvBX,C;6G A0BA,yB;MAAA,gD;MAAA,gE;MATnKI,0D;MASnKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OA AO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB92RO,W;QqB+2RP,kBAA kB,O;QACJ,OAzKE,YAAR,iBAAQ,C;QAyoKF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAA V,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M;O;KAvB X,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAxoKI,0D;MAwoKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe, OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBx4RO,W;QqBy4 RP,kBAakB,O;QACJ,OA3pKE,YAAR,iBAAQ,C;QA2pKF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UA AU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEEX,OAAO,M ;O;KAvBX,C;mGA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAA O,W;QACtB,sBAakB,sBAAK,CAAL,CAAIb,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;Q AA3C,arB16RO,W;QqBm6Re,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL ,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;M AAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAakB,sBAAK,CAAL,CAAIb,C;QAC oC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB37RO,W;QqB47Re,qB;QAAtB,iBAAc,CAAd, wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEEX,OAA O,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,O AAO,W;QACtB,sBAakB,sBAAK,CAAL,CAAIb,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C; QAA5C,arBp9RO,W;QqBq9Re,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAA L,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE; MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAakB,sBAAK,CAAL,CAAIb,C;QA

CqC,kBAAxB,eAAkB,cAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arB7+RO,W;QqB8+Re,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;QAEX,OAAO,M;O;KAtBX,C;iHAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,arBvgSO,W;QqBwgSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KA AV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arBjiSO,W;QqBkiSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KA AV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB3jSO,W;QqB4jSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KA AV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,cAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arBrlSO,W;QqBslSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KA AV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;QAEX,OAAO,M;O;KAvBX,C;iFA0BA,yB;MAxZA,gD;MAAA,gE;MAwZA,gD;QAgBW,sB;;UA tZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAYZH,OA zZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAwZzB,OA xZyB,C;UAA5C,arBrtRO,W;UqBstRP,kBAuZmB,O;UA tZH,2B;UAAhB,OA AgB,cAAhB,C;YAAgB,yB;YACZ,cAqZwB,SA rZV,CAAU,WAAV,EAAuB,OA AvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAkZP,yB;O;KAhBJ,C;iFAmBA,yB;MAIZA,gD;MAAA,gE;MAkZA,gD;QAgBW,sB;;UAhZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAmZH,OAnZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAkZzB,OAIzYB,C;UAA5C,arB9uRO,W;UqB+uRP,kBAiZmB,O;UAhZH,2B;UAAhB,OA AgB,cAAhB,C;YAAgB,yB;YACZ,cA+YwB,SA/YV,CAAU,WAAV,EAAuB,OA AvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA4YP,yB;O;KAhBJ,C;iFAmBA,yB;MA5YA,gD;MAAA,gE;MA4YA,gD;QAgBW,sB;;UA1YS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OA6YH,OA7YG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA4YzB,OA5YyB,C;UAA5C,arBvwRO,W;UqBwwRP,kBA2YmB,O;UA1YH,2B;UAAhB,OA AgB,cAAhB,C;YAAgB,yB;YACZ,cAyYwB,SAzYV,CAAU,WAAV,EAAuB,OA AvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAsYP,yB;O;KAhBJ,C;iFAmBA,yB;MA tYA,gD;MAAA,gE;MA sYA,gD;QAgBW,sB;;UApYS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAUyH,OA vYG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA sYzB,OA tYyB,C;UAA5C,arBhyRO,W;UqBiyRP,kBAqYmB,O;UApYH,2B;UAAhB,OA AgB,cAAhB,C;YAAgB,yB;YACZ,cAmYwB,SA nYV,CAAU,WAAV,EAAuB,OA AvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAgYP,yB;O;KAhBJ,C;+FAmBA,yB;MAhYA,gD;MAAA,gE;MAllKI,0D;MAk9KJ,gD;QAI BW,6B;;UA9XO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAiYI,OAjYJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAgYIB,OA hYkB,C;UAA5C,arB1zRO,W;UqB2zRP,kBA+X0B,O;UA9XZ,OA rMKE,YAAR,iBAAQ,C;UAqmKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA6X+B,SA7XjB,CAAU,KA AV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA0XP,gC;O;KAjBJ,C;+FAoBA,yB;MA1XA,gD;MAAA,gE;MApMKI,0D;MA89KJ,gD;QAI BW,6B;;UAxXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA2XI,OA3XJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA0XIB,OA1XkB,C;UAA5C,arBp1RO,W;UqBq1RP,kBAyX0B,O;UAxXZ,OA vnKE,YAAR,iBAAQ,C;UAunKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAuX+B,SA vXjB,CAAU,KA AV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAoXP,gC;O;KAjBJ,C;+FAoBA,yB;MApXA,gD;MAAA,gE;MAtnKI,0D;MA0+KJ,gD;QAI BW,6B;;UAIXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAqXIOArXJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAoXIB,OA pXkB,C;UAA5C,arB92RO,W;UqB+2RP,kBA mX0B,O;UAIXZ,OA zoKE,YAAR,iBAAQ,C;UAyoKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAiX+B,SAjXjB,CAAU,KA AV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA8WP,gC;O;KAjBJ,C;+FAoBA,yB;MA9WA,gD;MAAA,gE;MAxoKI,0D;MA s/KJ,gD;QAI BW,6B;;UA5

WO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA+WI,OA/WJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,C  
AAP,IAAb,C;UAA+B,sBA8WIB,OA9WkB,C;UAA5C,arBx4RO,W;UqBy4RP,kBA6W0B,O;UA5WZ,OA3pKE,Y  
AAR,iBAAQ,C;UA2pKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA2W+B,SA3WjB,CAAU,KAAV,EAAiB,  
WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAwwP,gC;O;  
KAjBJ,C;mFAoBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAh  
B,C;UAAgB,yB;UACZ,MnCvWsiD,SmCuwSjD,GnCvW2D,KAAK,GmCuwSzD,SAAS,OAAT,CnCvWSoE,KAA  
X,IAAf,C;;QmCyySrD,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W  
;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCvxSiD,SmCuxSjD,GnCvx2D,KAAK,GmCuxSz  
D,SAAS,OAAT,CnCvxSoE,KAAK,IAAf,C;;QmCyxSrD,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,wB;MAAA,s  
C;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCvySiD,SmCuySj  
D,GnCvy2D,KAAK,GmCuySzD,SAAS,OAAT,CnCvySoE,KAAK,IAAf,C;;QmCyySrD,OAAO,G;O;KAbX,C;m  
FAgBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB  
,yB;UACZ,MnCvzSiD,SmCuzSjD,GnCvz2D,KAAK,GmCuzSzD,SAAS,OAAT,CnCvzSoE,KAAK,IAAf,C;;QmC  
yzSrD,OAAO,G;O;KAbX,C;8FAgBA,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;  
QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MA  
CF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;  
MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,  
C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAg  
B,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;  
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYo  
B,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAE  
X,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q  
ACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAh  
B,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;M  
ADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,  
OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ  
,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB  
,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q  
;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MA  
EJ,OAAO,G;K;mFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,  
cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MAAA,S  
AWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,S  
AAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QA  
DhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,  
G;O;KAFx,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,c  
AAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MnC5xSA,  
6B;MmC4xSA,sC;QAaoB,Q;QADhB,UnC9xSmC,cmC8xSnB,CnC9xSmB,C;QmC+xSnB,2B;QAAhB,OAAgB,cA  
AhB,C;UAAgB,yB;UACZ,MnCImTiD,cmCkmTjD,GnCImT2D,KAAK,GmCkmTzD,SAAS,OAAT,CnCImToE,K  
AAX,IAAf,C;;QmComTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnC/ySA,6B;MmC+ySA,sC;QAaoB,Q;QADhB  
,UnCjzSmC,cmCizSnB,CnCjzSmB,C;QmCkzSnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCrmTiD,c  
mCqnTjD,GnCrmT2D,KAAK,GmCqnTzD,SAAS,OAAT,CnCrmToE,KAAK,IAAf,C;;QmCunTrD,OAAO,G;O;KA  
hBX,C;mFAmBA,yB;MnC10SA,6B;MmCk0SA,sC;QAaoB,Q;QADhB,UnCp0SmC,cmCo0SnB,CnCp0SmB,C;Qm  
Cq0SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCxoTiD,cmCwoTjD,GnCxoT2D,KAAK,GmCwoT  
zD,SAAS,OAAT,CnCxoToE,KAAK,IAAf,C;;QmC0oTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnC1SA,6B;M  
mCq1SA,sC;QAaoB,Q;QADhB,UnCv1SmC,cmCu1SnB,CnCv1SmB,C;QmCw1SnB,2B;QAAhB,OAAgB,cAAhB,  
C;UAAgB,yB;UACZ,MnC3pTiD,cmC2pTjD,GnC3pT2D,KAAK,GmC2pTzD,SAAS,OAAT,CnC3pToE,KAAK,IA  
Af,C;;QmC6pTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnBr2SA,+B;MmBq2SA,sC;QAaoB,Q;QADhB,UnBt2S  
qC,eAAW,oBmBs2S/B,CnBt2S+B,CAAX,C;QmBu2SrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnB

3qTmD,emB2qTnD,GnB3qT8D,KAAK,KmB2qT5D,SAAS,OAAT,CnB3qTuE,KAAH,CAAhB,C;;QmB6qTvD,O  
AAO,G;O;KAhBX,C;mFamBA,yB;MnBx3SA,+B;MmBw3SA,sC;QAaoB,Q;QADhB,UnBz3SqC,eAAW,oBmBy3  
S/B,CnBz3S+B,CAAX,C;QmB03SrB,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnB9rTmD,emB8rTnD,  
GnB9rT8D,KAAK,KmB8rT5D,SAAS,OAAT,CnB9rTuE,KAAH,CAAhB,C;;QmBgsTvD,OAAO,G;O;KAhBX,C;  
mFamBA,yB;MnB34SA,+B;MmB24SA,sC;QAaoB,Q;QADhB,UnB54SqC,eAAW,oBmB44S/B,CnB54S+B,CAA  
X,C;QmB64SrB,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBjtTmD,emBitTnD,GnBjtT8D,KAAK,Km  
BitT5D,SAAS,OAAT,CnBjtTuE,KAAH,CAAhB,C;;QmBmtTvD,OAAO,G;O;KAhBX,C;mFamBA,yB;MnB95SA  
,+B;MmB85SA,sC;QAaoB,Q;QADhB,UnB/5SqC,eAAW,oBmB+5S/B,CnB/5S+B,CAAX,C;QmBg6SrB,2B;QAaH  
B,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBpuTmD,emBouTnD,GnBpuT8D,KAAK,KmBouT5D,SAAS,OAAT,  
CnBpuTuE,KAAH,CAAhB,C;;QmBsuTvD,OAAO,G;O;KAhBX,C;IAmBA,kC;MA2DI,WpBnnTO,MAAO,KoBm  
nTG,cpBnnTH,EoBikTH,KAkDkB,OpBnnTf,C;MoBonTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB  
,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CARDO,EAAnB,KAqDqB,CAAM,CAAN,CARDF,CAq  
DrB,C;;MArDT,OAuDO,I;K;IApDX,kC;MAkEI,WpBtoTO,MAAO,KoBsoTG,cpBtoTH,EoB6kTH,KAyDkB,OpBt  
oTf,C;MoBuoTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4D  
P,sBAAK,CAAL,CA5DO,EAAnB,KA4DqB,CAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC  
;MAyEI,WpBzpTO,MAAO,KoBypTG,cpBzpTH,EoBylTH,KAgEkB,OpBzpTf,C;MoB0pTd,WAAW,iBAaA,IAAb  
,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAnB,KA  
mEqB,CAAM,CAAN,CAnEF,CAmErB,C;;MANET,OAqEO,I;K;IALEX,kC;MAGFI,WpB5qTO,MAAO,KoB4qTG,  
cpB5qTH,EoBqmTH,KAuEkB,OpB5qTf,C;MoB6qTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IA  
AIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAnB,KA0EqB,CAAM,CAAN,CA1EF,CA0ErB  
,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB9mTA,iB;MoB8mTA,8C;QAWI,WpBnnTO,MAAO,KoBm  
nTG,cpBnnTH,EoBmnTS,KAAM,OpBnnTf,C;QoBonTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IA  
AAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,O  
AAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBjoTA,iB;MoBioTA,8C;QAWI,WpBtoTO,MAAO,KoBsoTG,cp  
BtoTH,EoBsoTS,KAAM,OpBtoTf,C;QoBuoTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;U  
ACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;K  
AhBX,C;+EAmBA,yB;MAAA,gE;MpBppTA,iB;MoBopTA,8C;QAWI,WpBzpTO,MAAO,KoBypTG,cpBzpTH,E  
oBypTS,KAAM,OpBzpTf,C;QoB0pTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IA  
AK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,  
C;+EAmBA,yB;MAAA,gE;MpBvqTA,iB;MoBuqTA,8C;QAWI,WpB5qTO,MAAO,KoB4qTG,cpB5qTH,EoB4qT  
S,KAAM,OpB5qTf,C;QoB6qTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,W  
AAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAm  
BA,kC;MA8DoB,gB;MAHhB,gBAAgB,c;MACHB,WAAW,iBpBhvTJ,MAAO,KoBgvTsB,wBAnDzB,KAmDyB,E  
AAwB,EAAXB,CpBhvTtB,EoBgvTmD,SpBhvTnD,CoBgvTH,C;MACX,QAAQ,C;MACQ,OArDL,KAqDK,W;M  
AAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAvDqB,GAuDP,  
uBAAK,UAAAL,EAak,kBAAL,UAvDO,EAuDI,OAuDJ,CAuDrB,C;;MAvDT,OAyDO,I;K;IAtdX,kC;MAuEoB,g  
B;MAHhB,gBAAgB,c;MACHB,WAAW,iBpBrwTJ,MAAO,KoBqwTsB,wBA5DzB,KA4DyB,EAawB,EAAXB,Cp  
BrwTtB,EoBqwTmD,SpBrwTnD,CoBqwTH,C;MACX,QAAQ,C;MACQ,OA9DL,KA8DK,W;MAAhB,OAAgB,c  
AAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhEqB,GAgEP,uBAAK,UAAAL,E  
AAK,kBAAL,UAhEO,EAgEI,OAHEJ,CAGErB,C;;MAhET,OAkEO,I;K;IA/DX,kC;MAGFoB,gB;MAHhB,gBAAG  
B,c;MACHB,WAAW,iBpB1xTJ,MAAO,KoB0xTsB,wBArEzB,KAqEyB,EAawB,EAAXB,CpB1xTtB,EoB0xTmD,  
SpB1xTnD,CoB0xTH,C;MACX,QAAQ,C;MACQ,OA9EL,KAuEK,W;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;Q  
ACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAZEqB,GAYEP,uBAAK,UAAAL,EAak,kBAAL,UAEo,  
EAYEI,OAzeJ,CAYErB,C;;MAzET,OA2EO,I;K;IAxEX,kC;MAyFoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iB  
pB/yTJ,MAAO,KoB+yTsB,wBA9EzB,KA8EyB,EAawB,EAAXB,CpB/yTtB,EoB+yTmD,SpB/yTnD,CoB+yTH,C;  
MACX,QAAQ,C;MACQ,OA9FL,KA9FK,W;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,  
C;UAAoB,K;QACpB,IAAK,WAlFqB,GakFP,uBAAK,UAAAL,EAak,kBAAL,UAlFO,EakFI,OAIFJ,CakFrB,C;;  
MAIFT,OAoFO,I;K;+EAjFX,yB;MAAA,kF;MAAA,gE;MpB1uTA,iB;MoB0uTA,8C;QAcOB,UAEY,M;QAL5B,g

BAAgB,c;QACHB,WAAW,epBhvTJ,MAAO,KoBgvTsB,wBAAN,KAAM,EAAwB,EAAXB,CpBhvTtB,EoBgvTmD,SpBhvTnD,CoBgvTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpB/vTA,iB;MoB+vTA,8C;QAcOB,UAEY,M;QAL5B,gBAAgB,c;QACHB,WAAW,epBrwTJ,MAAO,KoBqwTsB,wBAAN,KAAM,EAAwB,EAAXB,CpBrwTtB,EoBqwTmD,SpBrwTnD,CoBqwTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpBpxTA,iB;MoBoxTA,8C;QAcOB,UAEY,M;QAL5B,gBAAgB,c;QACHB,WAAW,epB1xTJ,MAAO,KoB0xTsB,wBAAN,KAAM,EAAwB,EAAXB,CpB1xTtB,EoB0xTmD,SpB1xTnD,CoB0xTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;8EAqBA,yB;MAAA,kF;MAAA,gE;MpBzyTA,iB;MoByyTA,8C;QAcOB,UAEY,M;QAL5B,gBAAgB,c;QACHB,WAAW,epB/yTJ,MAAO,KoB+yTsB,wBAAN,KAAM,EAAwB,EAAXB,CpB/yTtB,EoB+yTmD,SpB/yTnD,CoB+yTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;IAqBA,kC;MA2DI,WpBn3TO,MAAO,KoBm3TG,cpBn3TH,EoBi0TH,KAkDkB,KpBn3Tf,C;MoBo3Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CARDO,EAAAnB,KAqDqB,aAAM,CAAN,CARDF,CAqDrB,C;;MARDT,OAuDO,I;K;IAPDX,kC;MAkEI,WpBt4TO,MAAO,KoBs4TG,cpBt4TH,EoB60TH,KAyDkB,KpBt4Tf,C;MoBu4Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAAnB,KA4DqB,aAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpBz5TO,MAAO,KoBy5TG,cpBz5TH,EoBy1TH,KAgEkB,KpBz5Tf,C;MoB05Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAAnB,KAmEqB,aAAM,CAAN,CAnEF,CAMerB,C;;MANET,OAqEO,I;K;IAIEX,kC;MAGFI,WpB56TO,MAAO,KoB46TG,cpB56TH,EoBq2TH,KAuEkB,KpB56Tf,C;MoB66Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAAnB,KA0EqB,aAAM,CAAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB92TA,iB;MoB82TA,8C;QAWI,WpBn3TO,MAAO,KoBm3TG,cpBn3TH,EoBm3TS,KAAM,KpBn3Tf,C;QoBo3Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBj4TA,iB;MoBi4TA,8C;QAWI,WpBt4TO,MAAO,KoBs4TG,cpBt4TH,EoBs4TS,KAAM,KpBt4Tf,C;QoBu4Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBv6TA,iB;MoBu6TA,8C;QAWI,WpB56TO,MAAO,KoB46TG,cpB56TH,EoB46TS,KAAM,KpB56Tf,C;QoB66Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAmBA,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnCjnUiD,SmCinUjD,GnCjnU2D,KAAK,GmCinUzD,OnCjnUoE,KAAAX,IAAf,C;;MmCmnUrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACjB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnB5nUmD,UmB4nUnD,GnB5nU8D,KAAK,KmB4nU5D,OnB5nUuE,KAAAX,CAAhB,C;;MmB8nUvD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnC7oUiD,SmC6oUjD,GnC7oU2D,KAAK,GAAW,CD2O5C,SoCk6TxB,OpCl6TkC,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MmC+oUrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnC3pUiD,SmC2pUjD,GnC3pU2D,KAAK,GAAW,CC4O5C,SkC+6TxB,OI C/6TkC,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MmC6pUrD,OAAO,G;K;+EAGX,yB;MAAA,0C;MnC x2TA,6B;MmCw2TA,4B;QAOI,OnCr2TmC,cmCq2TpB,IAAR,iBAAQ,CnCr2ToB,C;O;KmC81TvC,C;+EAUA,y

B;MAAA,0C;MnBn2TA,+B;MmBm2TA,4B;QAOI,OnBh2TsC,emBg2TvB,IAAR,iBAAQ,CnBh2TuB,C;O;KmBy  
1T1C,C;+EAUA,yB;MAAA,sC;MnC53TA,6B;MmC43TA,iBAOiB,yB;QpCz9Tb,6B;eoCy9Ta,c;UAAE,OpCh9To  
B,coCg9TpB,EpCh9T8B,KAAL,GAAiB,GAAtB,C;S;OoCg9TtB,C;MAPjB,4B;QA7iBoB,Q;QADhB,UnCp0SmC,  
cmCo0SnB,CnCP0SmB,C;QmCq0SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCxoTiD,cmCwoTjD,  
GnCxoT2D,KAAK,GAAW,CD2O5C,coC65Sf,OpC75SyB,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;Qm  
C2rUrD,OAjJBO,G;O;KA0iBX,C;+EAUA,yB;MAAA,sC;Mnct4TA,6B;MmCs4TA,iBAOiB,yB;QlCl+Tb,6B;ekCk  
+Ta,c;UAAE,OICz9ToB,ckCy9TpB,ElCz9T8B,KAAL,GAAiB,KAAtB,C;S;OkCy9TtB,C;MAPjB,4B;QApiBoB,Q;  
QADhB,UnCv1SmC,cmCu1SnB,CnCV1SmB,C;QmCw1SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,M  
nC3pTiD,cmC2pTjD,GnC3pT2D,KAAK,GAAW,CC4O5C,ckC+6Sf,OIC/6SyB,KAAL,GAAiB,KAAtB,CD5O4C,  
MAAX,IAAf,C;;QmCqsUrD,OAXiBO,G;O;KaiiBX,C;IC3vUA,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cA  
AX,C;MACb,YAA Y,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,  
WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,kC;MAQoB,UACL,M;MAHX,aAAa,eAAU,cAAV,C;MACb,YAA  
Y,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;  
;MACJ,OAAO,M;K;IAGX,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAA Y,C;MACI,2B;M  
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M  
;K;IAGX,oC;MAQoB,UACL,M;MAHX,aAAa,iBAA Y,cAAZ,C;MACb,YAA Y,C;MACI,2B;MAAhB,OAAgB,cA  
AhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,2B;MA  
QoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MpCAiD,SoCAjD,GpCA  
2D,KAAK,GoCAzD,OpCAoE,KAAX,IAAf,C;;MoCERD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;  
MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MpBXmD,UoBwN D,GpBX8D,KAAK,KoBW5D,OpB  
XuE,KAAX,CAAhB,C;;MoBavD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OA  
AgB,cAAhB,C;QAAgB,yB;QACZ,MpC5BiD,SoC4BjD,GpC5B2D,KAAK,GAAW,CD2O5C,SqC/MxB,OrC+MkC  
,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MoC8BrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB  
,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MpC1CiD,SoC0CjD,GpC1C2D,KAAK,GAAW,CC  
4O5C,SmCIMxB,OnCkMkC,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MoC4CrD,OAAO,G;K;IC3GX,w  
B;MAMI,OrCuCkE,YqCvCvD,CrCuCwE,KAAjB,EqCvCiD,CrCuC+E,KAA7B,CqCvCvD,KAAJ,GAAY,CAAZ,G  
AAmB,C;K;IAG9B,wB;MAMI,OrBsCmE,aqBtCxD,CrBsC0E,KAAiB,EqBtCnD,CrBsCiF,KAA9B,CqBtCxD,KA  
AJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OtCKgF,0BsCLrE,CtCgP2B,KAAL,GAAiB,GA3O8B,EsCLh  
E,CtCgPsB,KAAL,GAAiB,GA3O8B,CsCLrE,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OpClfI,0Bo  
CjTE,CpCwO2B,KAAL,GAAiB,KApO+B,EoCjJE,CpCwOsB,KAAL,GAAiB,KApO+B,CoCjTE,KAAJ,GAAY,CA  
AZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAA  
S,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,C  
AAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,C  
AAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB  
;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAG  
X,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,  
EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C  
;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,U  
AAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACv  
B,OAAO,G;K;IAGX,wB;MAMI,OrCjFkE,YqCiFvD,CrCjFwE,KAAjB,EqCiFiD,CrCjF+E,KAA7B,CqCiFvD,KAA  
J,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBIFmE,aqBkFxD,CrBIF0E,KAAiB,EqBkFnD,CrBIFiF,KAA9  
B,CqBkFxD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OtCnHgF,0BsCmHrE,CtCwH2B,KAAL,GAA  
iB,GA3O8B,EsCmHhE,CtCwHsB,KAAL,GAAiB,GA3O8B,CsCmHrE,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B  
,wB;MAMI,OpCpHiF,0BoCoHtE,CpCgH2B,KAAL,GAAiB,KApO+B,EoCoHjE,CpCgHsB,KAAL,GAAiB,KApO  
+B,CoCoHtE,KAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CA  
AN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,  
MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QA



OI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAA X,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB; QAAO,MAAM,SAAM,GAAN,EAAW,CAA X,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MA CA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAA X,C;;MACvB,OAAO,G; K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM, GAAN,EAAW,CAA X,C;;MACvB,OAAO,G;K;gFC7OX,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kB AAO,cAAP,C;O;KATX,C;gFAYA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KAT X,C;;IAYA,sC;;QASQ,OAAc,WAAP,MAAO,EAAS,SAAT,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAE,QAAz B,C;;UAHV,O;;K;IAOJ,sC;;QASQ,OAAc,YAAP,MAAO,EAAU,SAAV,C;;QACHB,+C;UACE,MAAM,2BAAuB, CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O; KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;M AMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,WAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C;MAMI,IAAI,m BAAJ,C;QACI,OAAO,I;MACX,OAAc,YAAP,MAAO,EAAU,SAAV,C;K;oFAGIB,8B;MASI,OAAO,WAAW,IA AX,IAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MASI,OAAO,WAAW,IAAX,IAAmB,2BAAS,OAAT,C;K;IAG9B,u C;MAMI,OAAO,2BvC4K4B,SuC5KnB,KvC4K6B,KAAL,GAAiB,GAAtB,CuC5K5B,C;K;IAGX,uC;MAMI,OAA O,2BvC6K8B,UAAW,oBuC7KhC,KvC6K2B,KAAC,CAAL,UAAAN,CuC7K9B,C;K;IAGX,uC;MAMI,OAAO,2Bt CwL8B,UAAW,oBsCxLhC,KtCwL2B,KAAC,CAAL,iBAAN,CsCxL9B,C;K;IAGX,uC;MAMY,Q;MAAD,cAAC, OtBqF4C,UsBrF5C,KtBqFkD,yBsBrFxC,EtBqFwC,CAAN,CsBrF7C,wBAA8B,2BAA9B,Q;MAAA,W;QAAqC,o CtCoPR,SsCpPiB,KtB6KIB,KhBuEW,QAAV,CsCpPQ,C;;MAA5C,a;K;IAGJ,uC;MAMI,OAAO,2BrCyI4B,SqCzI nB,KrCyI6B,KAAL,GAAiB,KAAtB,CqCzI5B,C;K;IAGX,uC;MAMI,OAAO,2BrC0I8B,UAAW,oBqC1IhC,KrC0I 2B,KAAC,CAAL,YAAN,CqC1I9B,C;K;IAGX,kC;MASI,OAAO,uCAAgB,yBvCmHY,SuCnHI,SvCmHM,KAAL, GAAiB,GAAtB,CuCnHZ,EvCmHY,SuCnHmB,EvCmHT,KAAL,GAAiB,GAAtB,CuCnHZ,EAA4C,EAA5C,C;K;I AG3B,kC;MASI,OAAO,uCAAgB,yBAAGB,SAAhB,EAA5B,EAAtB,EAA0B,EAA1B,C;K;IAG3B,kC;MASI,OAA O,wCAAiB,yBAAGB,SAAhB,EAA5B,EAAtB,M;K;IAG5B,kC;MASI,OAAO,uCAAgB,yBrCgFY,SqChFI,SrCgF M,KAAL,GAAiB,KAAtB,CqChFZ,ErCgFY,SqChFmB,ErCgFT,KAAL,GAAiB,KAAtB,CqChFZ,EAA4C,EAA5C, C;K;IAG3B,gC;MAMI,OAAO,uCAAgB,yBAAGB,cAAhB,EAA5B,eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IA G3B,gC;MAMI,OAAO,wCAAiB,yBAAGB,cAAhB,EAA5B,eAAtB,EAA8B,cAAD,aAA7B,C;K;IAG5B,iC;MAMI, oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,S AAK,KAAL,GAAy,CAAhB,GAAMB,IAANB,GAA6B,CAAC,IAAD,IAA1D,C;K;IAG3B,iC;MAMI,oBAAoB,kB AAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,wCAAiB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAA L,cAAy,CAAhB,GAAMB,IAANB,GAA8B,IAAD,aAA1D,C;K;IAG5B,iC;MAQI,IvC/OgF,0BuC+O5E,EvCJkC,K AAL,GAAiB,GA3O8B,EUc+Ote,6BAAM,UvCJsB,KAAL,GAAiB,GA3O8B,CuC+O5E,KAAJ,C;QAA2B,OAAO, iCAAU,M;MACHC,WvC6BuB,SuC7B5B,SvC6BsC,KAAL,GAAiB,GAAtB,C;MuC7BV,YAAK,W;MAA9B,OtCj D6D,oBAhJP,SAAU,CD8N7B,SuC7BV,EvC6BoB,KAAL,GAAiB,GAAtB,CC9N6B,MAAK,GDAK,KCAO,KAA Z,IAAf,CAGJO,C;K;IsCoDjE,iC;MAQI,ItC3OkE,YsC2O9D,EtC3O+E,KAAjB,EsC2OxD,4BAAK,UtC3OgF,KAA 7B,CsC2O9D,KAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OtC7D6D,csC6DtD,StC7DsD,EAhJP,SsC6MtC,EtC7 MgD,KAAC,GAAy,CsC6M5D,WtC7M4D,MAAZ,IAAf,CAGJO,C;K;IsCgEjE,iC;MAQI,ItB/OmE,asB+O/D,EtB/ OiF,KAAIB,EsB+OzD,6BAAM,UtB/OiF,KAA9B,CsB+O/D,KAAJ,C;QAA2B,OAAO,kCAAW,M;MAC7C,OtBzE +D,iBsByExD,StBzEwD,EA7IP,UsBsNx,C,EtBtNmD,KAAC,UAAy,ChByP/C,UAAW,oBAAL,CsCnCb,WtCmC sB,MAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,CA6IO,C;K;IsB4EnE,iC;MAQI,IrC3QiF,0BqC2Q7E,ErCvC kC,KAAL,GAAiB,KApO+B,EqC2QvE,8BAAO,UrCvCqB,KAAL,GAAiB,KApO+B,CqC2Q7E,KAAJ,C;QAA4B, OAAO,iCAAU,M;MACjC,WrCNuB,SqCM5B,SrCnS,C,KAAL,GAAiB,KAAtB,C;MqCMV,YAAK,W;MAA9B,Ot CrF6D,oBAhJP,SAAU,CC+N7B,SqCMV,ErCNoB,KAAL,GAAiB,KAAtB,CD/N6B,MAAK,GCAK,KDAO,KAA Z,IAAf,CAGJO,C;K;IsCwFjE,kD;MAUI,OtCjRkE,YsCiRvD,StCjRwE,KAAjB,EsCiRhD,YtCjR6E,KAA7B,CsCiR vD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OtBtRmE,asBsRxD,StBtR0E,KAAIB,EsBsRjD,YtBtR+ E,KAA9B,CsBsRxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OvC3TgF,0BuC2TrE,SvChF2B,KAA

L,GAAiB,GA3O8B,EuC2T9D,YvChFoB,KAAL,GAAiB,GA3O8B,CuC2TrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K; IAGtD,kD;MAUI,OrChUiF,0BqCgUtE,SrC5F2B,KAAL,GAAiB,KApO+B,EqCgU/D,YrC5FoB,KAAL,GAAiB,K ApO+B,CqCgUtE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtCrUkE,YsCqUvD,StCrUwE,KAAjB,E sCqUhD,YtCrU6E,KAA7B,CsCqUvD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtB1UmE,asB0UxD, StB1U0E,KAAiB,EsB0UjD,YtB1U+E,KAA9B,CsB0UxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,O vC/WgF,0BuC+WrE,SvCpI2B,KAAL,GAAiB,GA3O8B,EuC+W9D,YvCpIoB,KAAL,GAAiB,GA3O8B,CuC+Wr E,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OrCpXiF,0BqCoXtE,SrChJ2B,KAAL,GAAiB,KApO+B,E qCoX/D,YrChJoB,KAAL,GAAiB,KApO+B,CqCoXtE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,4D;MAUI,ItCz XkE,YsCyX9D,YtCzX+E,KAAjB,EsCyX/C,YtCzX4E,KAA7B,CsCyX9D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDA AiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,ItC1XkE,YsC0X9D,StC1X+E,KAAjB,EsC0XvD,YtC1XoF,K AA7B,CsC0X9D,IAAJ,C;QAAyB,OAAO,Y;MACHC,ItC3XkE,YsC2X9D,StC3X+E,KAAjB,EsC2XvD,YtC3XoF, KAA7B,CsC2X9D,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,4D;MAUI,ItBjYmE,asBiY/D,YtBjYiF, KAAiB,EsBiYhD,YtBjY8E,KAA9B,CsBiY/D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAA pF,MAAzB,C;MACvC,ItBIYmE,asBkY/D,StBIYiF,KAAiB,EsBkYxD,YtBIYsF,KAA9B,CsBkY/D,IAAJ,C;QAAy B,OAAO,Y;MACHC,ItBnYmE,asBmY/D,StBnYiF,KAAiB,EsBmYxD,YtBnYsF,KAA9B,CsBmY/D,IAAJ,C;QAA yB,OAAO,Y;MACHC,OAAO,S;K;IAGX,4D;MAUI,IvCzagF,0BuCya5E,YvC9LkC,KAAL,GAAiB,GA3O8B,EuC ya7D,YvC9LmB,KAAL,GAAiB,GA3O8B,CuCya5E,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAo F,YAApF,MAAzB,C;MACvC,IvC1agF,0BuC0a5E,SvC/LkC,KAAL,GAAiB,GA3O8B,EuC0arE,YvC/L2B,KAAL, GAAiB,GA3O8B,CuC0a5E,IAAJ,C;QAAyB,OAAO,Y;MACHC,IvC3agF,0BuC2a5E,SvChMkC,KAAL,GAAiB,G A3O8B,EuC2arE,YvChM2B,KAAL,GAAiB,GA3O8B,CuC2a5E,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;I AGX,4D;MAUI,IrCjbiF,0BqCib7E,YrC7MkC,KAAL,GAAiB,KApO+B,EqCib9D,YrC7MmB,KAAL,GAAiB,KAp O+B,CqCib7E,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IrClibF, 0BqCkb7E,SrC9MkC,KAAL,GAAiB,KApO+B,EqCkbtE,YrC9M2B,KAAL,GAAiB,KApO+B,CqCkb7E,IAAJ,C; QAAyB,OAAO,Y;MACHC,IrCnbiF,0BqCmb7E,SrC/MkC,KAAL,GAAiB,KApO+B,EqCmbtE,YrC/M2B,KAAL,G AAIb,KApO+B,CqCmb7E,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,uC;MAcW,Q;MAJP,IAAI,8CA AJ,C;QACI,OAAy,WAAL,SAAK,EAAe,KAAf,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4C AAyC,KAAzC,MAAzB,C;MAEvB,ItC9b8D,YsC8b9D,StC9b+E,KAAjB,EsC8bvD,KAAM,MtC9b8E,KAA7B,Cs C8b9D,K;QAA4B,OAAN,KAAM,M;;QAC5B,ItC/b8D,YsC+b9D,StC/b+E,KAAjB,EsC+bvD,KAAM,atC/b8E,KA A7B,CsC+b9D,K;UAAmC,OAAN,KAAM,a;;UAC3B,gB;;MAHZ,W;K;IAOJ,uC;MAcW,Q;MAJP,IAAI,8CAAJ,C ;QACI,OAAy,WAAL,SAAK,EAAgB,KAAhB,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4C AAyC,KAAzC,MAAzB,C;MAEvB,ItB3c+D,asB2c/D,StB3ciF,KAAiB,EsB2cxD,KAAM,MtB3cgF,KAA9B,CsB2c /D,K;QAA4B,OAAN,KAAM,M;;QAC5B,ItB5c+D,asB4c/D,StB5ciF,KAAiB,EsB4cxD,KAAM,atB5cgF,KAA9B, CsB4c/D,K;UAAmC,OAAN,KAAM,a;;UAC3B,gB;;MAHZ,W;K;IC/fJ,2B;MAUoB,Q;MADhB,UAAgB,W;MAC A,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MvCoDiD,SuCPDjD,GvCoD2D,KAAK,GuCPDzD,OvCoDo E,KAAx,IAAf,C;;MuClDrD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAiB,2B;MACD,2B;MAAhB,OAAgB, cAAhB,C;QAAgB,yB;QACZ,MvBuCmD,UuBvCnD,GvBuC8D,KAAK,KuBvC5D,OvBuCuE,KAAx,cAAhB,C;; MuBrCvD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB ,yB;QACZ,MvCoBiD,SuCPBjD,GvCoB2D,KAAK,GAAW,CD2O5C,SwC/PxB,OxC+PkC,KAAL,GAAiB,GAAtB, CC3O4C,MAAX,IAAf,C;;MuClBrD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB, OAAgB,cAAhB,C;QAAgB,yB;QACZ,MvClIiD,SuClJd,GvClI2D,KAAK,GAAW,CC4O5C,SsChPxB,OtCgPkC,KA AL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MuCFrD,OAAO,G;K;,,,,;ICuCP,iD;MAAA,qE;MAAgB,4B;MANp B,uC;MAMI,Y;K;IACA,4D;MAAA,qE;MAAgC,wBAAM,OAAN,Q;MAPpC,uC;MAOI,Y;K;IACA,mE;MAAA,q E;MAAmD,6BAAM,OAAN,EAAe,KAAf,C;MARvD,uC;MAQI,Y;K;IACA,0D;MAAA,qE;MAAiC,wBAAM,KA AN,Q;MATrC,uC;MASI,Y;K;ICxGJ,gC;K;,,,,;ICuBoC,wC;8BAAsC,O;K;,,,,,;yCC0RtE,6B;MA SI,MAAM,yB;K;,,,,;0CAyDV,sB;MASI,OAAO,I;K;,,,,,;,,,,,;ICnYf,wB;K;kCAEI,Y;MAA4B,sB;K;;IAMhC, wB;K;kCAEI,Y;MAA4B,mC;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,uB;K;iCAEI,Y;MAA4B,qB;K;;I AMhC,wB;K;kCAEI,Y;MAA4B,sB;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,0B;K;oCAEI,Y;MAA4B, wB;K;;IAMhC,2B;K;qCAEI,Y;MAA4B,yB;K;;ICtDM,oD;MAA2C,uB;MAAjB,gB;MAC5D,sBAAgC,InBkCU,I;

MmBjC1C,iBAAmC,YAAO,CAAX,GAAC,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAA5B,GAAqC,KnB  
gCK,ImBhC1C,GAAqD,mB;K;gDAErD,Y;MAAkC,qB;K;iDAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,UAAS,mBA  
Ab,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAGV,4BAAQ,SAAR,I;;MAEJ,OAAa,O  
AAN,KAAM,C;K;;IAQgB,mD;MAAyC,sB;MAAjB,gB;MACzD,sBAAgC,I;MAChC,iBAAmC,YAAO,CAAX,GA  
Ac,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAAJ,GAAa,KAAb,GAAwB,mB;K;+CAEhD,Y;MAAkC,qB;  
K;+CAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,UAAS,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QA  
C3B,iBAAU,K;;QAGV,4BAAQ,SAAR,I;;MAEJ,OAAO,K;K;;IAQuB,oD;MAA4C,uB;MAAIB,gB;MAC5D,sBAA  
iC,I;MACjC,iBAAmC,uBAAO,CAAX,GAAC,sBAAS,IAAT,MAAd,GAAiC,sBAAS,IAAT,M;MAChE,cAA6B,cA  
AJ,GAAa,KAAb,GAAwB,mB;K;gDAEjD,Y;MAAkC,qB;K;iDAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,cAAS,mBA  
AT,CAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAGV,8BAAQ,SAAR,C;;MAEJ,  
OAAO,K;K;;IC9DX,oD;MA6CA,uC;MAtCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAyB,wBAAzB,C;MAC5B,I  
AAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,wEAzB,C;MAG5C,aAGyB,K;MAEzB,YAGuF,OAA/D,0BAA0  
B,KpBcR,IoBdlB,EAAcC,YpBcpB,IoBdlB,EAAyD,IAAzD,CAA+D,C;MAEvF,YAGuB,I;K;yCAEvB,Y;MAAwC,  
mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC,OAAI,YAAO,CAAX,GAAC,aAA  
Q,SAATB,GAAgC,aAAQ,S;K;uCAE7E,iB;MACI,iDAA6B,kBAAa,KAAM,UAAAnB,KAC7B,eAAS,KAAM,MAAf  
,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KAD/B,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,  
EAAf,GAAwB,OAAM,OAAC,UpBRG,IoBQR,UAAkB,SpBRV,IoBQR,KAAN,SAAqC,SAArC,I;K;yCAE5B,Y;  
MAAkC,OAAI,YAAO,CAAX,GAAC,oBAAE,UAAF,+BAAU,SAAV,eAAqB,SAAnC,GAA8C,oBAAE,UAAF,qC  
AAgB,SAAhB,gBAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,UAAh  
B,EAA4B,QAA5B,EAAcC,IAAtC,C;K;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAiBA,mD;MA6CA,sC;  
MAtCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAyB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAA  
a,gCAAyB,wEAzB,C;MAG5C,aAGwB,K;MAExB,YAGuB,0BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C  
;MAEvB,YAGuB,I;K;wCAEvB,Y;MAAuC,kCAAuB,UAAvB,EAA8B,SAA9B,EAAoC,SAAPC,C;K;uCAEvC,Y;  
MAMqC,OAAI,YAAO,CAAX,GAAC,aAAQ,SAATB,GAAgC,aAAQ,S;K;sCAE7E,iB;MACI,gDAA4B,kBAAa,KA  
AM,UAAAnB,KAC5B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KADhC,CAA5  
B,C;K;wCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UAAAL,QAAa,SAAb,IAAN,SAA2B,S  
AA3B,I;K;wCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAGB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GAAgD,U  
AAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,oC;MAAA,wC;K;iEACI,sC;MAQwF,0BAAe,U  
AAF,EAA2B,QAA3B,EAAqC,IAArC,C;K;;IAT5F,gD;MAAA,+C;QAAA,8B;;MAAA,wC;K;;IAiBA,oD;MA6CA,  
uC;MAtCI,IAAI,gBAAJ,C;QAAgB,MAAa,gCAAyB,wBAAzB,C;MAC7B,IAAI,sCAAJ,C;QAA4B,MAAa,gCAAy  
B,yEAzB,C;MAG7C,aAGyB,K;MAEzB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,  
YAGwB,I;K;yCAExB,Y;MAAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC  
,OAAI,uBAAO,CAAX,GAAC,2BAAQ,SAAR,KAAAd,GAAgC,2BAAQ,SAAR,K;K;uCAErE,iB;MACI,iDAA6B,k  
BAAa,KAAM,UAAAnB,KAC7B,mBAAS,KAAM,MAAf,KAawB,kBAAQ,KAAM,KAAd,CAAxB,IAA8C,kBAA  
Q,KAAM,KAAd,CADjB,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,iCAAM,eAA  
W,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EAUV,CAAV,CAAPC,CAAN,MAAuE,cAAU,6BAA  
U,EAUV,CAAV,CAAvE,CAAiG,Q;K;yCAE7H,Y;MAAkC,OAAI,uBAAO,CAAX,GAAGB,UAAF,qBAAU,SA  
V,yBAAqB,SAArB,WAAAd,GAAgD,UAAF,2BAAgB,SAAhB,yBAA6B,SAAD,aAA5B,W;K;IAEHf,qC;MAAA,y  
C;K;kEACI,sC;MAQ4F,2BAAgB,UAAhB,EAA4B,QAA5B,EAAcC,IAAtC,C;K;;IAThG,iD;MAAA,gD;QAAA,+  
B;;MAAA,yC;K;;;6CCIKa,iB;MAGkD,+BAAS,UAAT,UAAkB,wBAAS,iBAAT,M;K;oCAEpE,Y;MAKgC,oCA  
AQ,iBAAR,K;K;;I7CpBd,wC;MAsBIB,iC;MAtBsD,2BAAgB,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFA  
C7B,Y;MAAQ,8B;K;yFACD,Y;MAAQ,6B;K;2CAExC,iB;MAA8C,qBAAS,KAAT,IAAkB,SAAS,S;K;kCAEzE,Y  
;MAKkC,oBAAQ,S;K;iCAE1C,iB;MACI,2CAAuB,kBAAa,KAAM,UAAAnB,KACvB,eAAS,KAAM,MAAf,IAAw  
B,cAAQ,KAAM,KADf,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAC,UwBkBS,IxBI  
Bd,UAAkB,SwBkJ,IxBIBd,K;K;mCAE5B,Y;MAAkC,2BAAE,UAAF,+BAAU,SAAV,C;K;IAEIC,+B;MAAA,m  
C;MACI,aAC8B,cAAY,OAAF,CAAE,CAAZ,EAAwB,OAAF,CAAE,CAAxB,C;K;;IAFIC,2C;MAAA,0C;QAAA,  
yB;;MAAA,mC;K;;IASiB,uC;MAsBjB,gC;MAtBmD,0BAAe,KAAf,EAAcB,YAAATB,EAAoC,CAAPC,C;K;iFAC3  
B,Y;MAAQ,iB;K;wFACD,Y;MAAQ,gB;K;0CAEvC,iB;MAA6C,qBAAS,KAAT,IAAkB,SAAS,S;K;iCAExE,Y;M

AKkC,oBAAQ,S;K;gCAE1C,iB;MACI,0CAAsB,kBAAa,KAAM,UAAAnB,KACtB,eAAS,KAAM,MAAf,IAAwB,c  
AAQ,KAAM,KADhB,CAAtB,C;K;kCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,UAAAL,QAAA,SA  
Ab,I;K;kCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,8B;MAAA,kC;MACI,aAC6B,aAAS,CAAT,EAA  
Y,CAAZ,C;K;;IAFjC,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;;IASkB,wC;MA5BlB,iC;MatBsD,2BAAgB,KAA  
hB,EAAuB,YAAvB,K;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;2CAExC,iB;MAA8C,kCAAS,KAAT,  
UAAkB,sBAAS,SAAT,M;K;kCAEhE,Y;MAKkC,kCAAQ,SAAR,K;K;iCAEIC,iB;MACI,2CAAuB,kBAAa,KAA  
M,UAAAnB,KACvB,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CADD,CAAvB,C;K;mCAGJ,Y;MA  
CI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,eAAW,8BAAW,EAAx,CAAX,CAAN,MAAoC,cAAU,6BAAU,E  
AAV,CAAV,CAAP,C,CAA8D,Q;K;mCAE1F,Y;MAAkC,OAAE,UAAF,qBAAU,SAAV,W;K;IAEIC,+B;MAAA,m  
C;MACI,aAC8B,qB;K;;IAFIC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;I8C9EJ,gB;MAAA,oB;K;8BAIL,Y;MA  
A0B,oB;K;;IAJ9B,4B;MAAA,2B;QAAA,U;;MAAA,oB;K;ICEA,yC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,uC  
;MAAA,0C;O;MAII,KE;MAEA,wF;MAEA,oF;MAEA,wE;MAEA,kE;MAEA,oF;MAEA,sF;MAEA,8E;MAEA,wE  
;MAEA,sF;MAEA,uF;MAEA,iE;MAEA,6E;MAEA,iE;MAEA,2E;K;;IA5BA,8C;MAAA,6B;MAAA,sC;K;;IAEA,  
yD;MAAA,6B;MAAA,iD;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,8C;M  
AAA,6B;MAAA,sC;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,oD;MAA  
A,6B;MAAA,4C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,wD;MAAA,6  
B;MAAA,gD;K;;IAEA,6C;MAAA,6B;MAAA,qC;K;;IAEA,mD;MAAA,6B;MAAA,2C;K;;IAEA,6C;MAAA,6B;  
MAAA,qC;K;;IAEA,kD;MAAA,6B;MAAA,0C;K;;IAhCJ,mC;MAAA,+oB;K;;IAAA,wC;MAAA,a;aAAA,O;UAA  
A,2C;aAAA,kB;UAAA,sD;aAAA,gB;UAAA,oD;aAAA,U;UAAA,8C;aAAA,O;UAAA,2C;aAAA,gB;UAAA,oD;a  
AAA,iB;UAAA,qD;aAAA,a;UAAA,iD;aAAA,U;UAAA,8C;aAAA,iB;UAAA,qD;aAAA,iB;UAAA,qD;aAAA,M;  
UAAA,0C;aAAA,Y;UAAA,gD;aAAA,M;UAAA,0C;aAAA,W;UAAA,+C;;UAAA,uE;;K;;IAqCA,4C;MAAA,e;M  
AAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAMI,0E;MAEA,0E;MAEA,4E;K;;IAJA,kD;MAAA,gC;MAAA,0  
C;K;;IAEA,kD;MAAA,gC;MAAA,0C;K;;IAEA,mD;MAAA,gC;MAAA,2C;K;;IAVJ,sC;MAAA,sI;K;;IAAA,2C;M  
AAA,a;aAAA,Q;UAAA,+C;aAAA,Q;UAAA,+C;aAAA,S;UAAA,gD;;UAAA,0E;;K;;IAwB8B,gC;MAAC,oC;K;;I  
AQE,0B;MAAC,qB;QAAA,iD;MAAA,kB;K;;IAEIC,sB;K;;IAMA,4B;K;;IC/EA,yB;K;;IAQA,6B;K;;ICnBA,mB;  
MAEI,UAAU,IAAI,C;MACd,OAAW,OAAO,CAAX,GAAc,GAAd,GAAuB,MAAM,CAAN,I;K;IAGIC,qB;MACI,  
UAAU,SAAI,CAAJ,C;MACV,OAAW,kBAAO,CAAX,GAAc,GAAd,GAAuB,QAAM,CAAN,C;K;IAGIC,mC;MA  
EI,OAAO,IAAI,IAAI,CAAJ,EAAO,CAAP,IAAY,IAAI,CAAJ,EAAO,CAAP,CAAZ,IAAJ,EAA2B,CAA3B,C;K;I  
AGX,qC;MACI,OAAO,MAAI,MAAI,CAAJ,EAAO,CAAP,WAAY,MAAI,CAAJ,EAAO,CAAP,CAAZ,CAAJ,EA  
A2B,CAA3B,C;K;IAGX,qD;MAkBI,WAAO,CAAP,C;QAD2E,OAC3D,SAAS,GAAb,GAakB,GAAIB,GAA2B,M  
AAM,iBAaIB,GAAjB,EAA5B,KAAtB,EAA6B,IAA7B,CAAN,I;WACvC,WAAO,CAAP,C;QAF2E,OAE3D,SA  
S,GAAb,GAakB,GAAIB,GAA2B,MAAM,iBAaIB,KAAjB,EAAwB,GAAxB,EAA6B,CAAC,IAAD,IAA7B,CAA  
N,I;;QAC/B,MAAa,gCAAyB,eAAzB,C;K;IAGzB,uD;MAkBI,sBAAO,CAAP,C;QAD+E,OAC/D,sBAAS,GAAT,  
MAAJ,GAakB,GAAIB,GAA2B,aAAM,mBAaIB,GAAjB,EAA5B,KAAtB,EAA6B,IAA7B,CAAN,C;WACvC,sB  
AAO,CAAP,C;QAF+E,OAE/D,sBAAS,GAAT,MAAJ,GAakB,GAAIB,GAA2B,QAAM,mBAaIB,KAAjB,EAAw  
B,GAAxB,EAA8B,IAAD,aAA7B,CAAN,C;;QAC/B,MAAa,gCAAyB,eAAzB,C;K;IC7DjB,kD;MAAA,8B;MACI,  
aAAy,C;K;oDACZ,Y;MAAyB,oBAAQ,gBAAl,O;K;iDACrC,Y;MAAgD,Q;MAA1B,IAAI,aAAQ,gBAAl,OAAhB  
,C;QAAA,OAA5B,iBAAl,iBAAJ,EAAI,yBAAJ,O;;QAAkB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAPhF,oC;M  
AEI,IAD8D,IAC9D,S;QACI,UAA0B,K;QAF0B,2C;;QAAA,QAAM,IAAN,C;eASxD,c;YATwD,OAStC,qBAAqB,  
KAARb,C;eACIB,W;YAVwD,OAUzC,kBAakB,KAAIB,C;eACf,Y;YAXwD,OAWxC,mBAAMb,KAAAnB,C;eAC  
hB,W;YAZwD,OAYzC,kBAakB,KAAIB,C;eACf,U;YAbwD,OAA1C,iBAaIB,KAAjB,C;eACd,W;YAdwD,OAcz  
C,kBAakB,KAAIB,C;eACf,Y;YAfWd,OAexC,mBAAMb,KAAAnB,C;eAchB,a;YAhBwD,OAgbvC,oBAAoB,KA  
ApB,C;;YACT,MAAM,6BAAsB,2DAA+C,IAA/C,CAAtB,C;;K;IAIuC,2D;MAAA,kC;MAAS,0B;MAC9D,aAAY,  
C;K;2DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;+DACvC,Y;MAA2D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;  
QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJnF,qC;  
MACyD,oD;K;IAON,wD;MAAA,kC;MAAS,uB;MACxD,aAAy,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;  
yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBA  
AN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOE,yD;MAAA,kC;MAAS,w



B;;;UAIZ,cAAc,oB;UACd,IAAI,YAAy,yBAAhB,C;YAAqC,M;UACrC,kBAAgB,O;UAChB,qBAAmB,I;;UAEnB ,kBAAgB,I;UAChB,qBAAmB,S;;QAGvB,gC;QAEA,IAAI,wCAAJ,C;UAEI,YAAU,U;;UAGV,U;UAAA,0C;YET hB,8BDgDQ,WAAO,qBAAP,CChDR,C;YFSgB,a;;YAAA,a;UAAA,mB;YAEK,UEpBrB,oBDgDQ,WD5B+B,eC4 B/B,CChDR,C;;UFqBgB,M;;;K;mDAMhB,Y;MACI,kBAAkB,mB;MACIB,IAAI,uBAAuB,gBAAGB,IAA3C,C;QA CI,uCAAQ,yCAAR,EAAmC,wCAA+B,WAA/B,C;;MAEvC,sBAAoB,mC;K;;IAM5B,iC;MAAA,qC;K;gGAEQ,Y; M7C0DyC,MAAM,6B6C1DjC,uC7C0D+D,WAA9B,C;K;yD6CxDnD,kB;M7CwD6C,MAAM,6B6CvDzC,uC7Cu DuE,WAA9B,C;K;+C6CpDnD,Y;MAAkC,8C;K;;IARiC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IGyDA,mG;IA AA,yH;IAAA,6F;MAKW,kC;MAAS,4C;K;IALpB,sEAMQ,Y;MACI,Q;MAAA,sC;QAAiB,U;;MACjB,OAAO,oB; K;IARnB,6G;sJAJIA,iC;MAGBU,OAAK,SAAL,CAAIb,UAAjB,EAA6B,KAA7B,C;K;wJAEV,2C;MAiBU,OAAK ,SAAL,CAAIb,QAAjB,EAA2B,UAA3B,EAAuC,KAAvC,C;K;wJAEV,kD;MAKU,OAAK,SAAL,CAAIb,QAAjB, EAA2B,KAA3B,EAAkC,UAAIC,EAA8C,KAA9C,C;K;IAGC6C,oG;MAAA,mB;QAC3C,OAAK,iCAAL,CAAIb,k BAAjB,C;O;K;IA/BZ,6D;MA0BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAIb,UAAjB,E AA6B,IAA7B,C;;QA8D0B,Q;QAhE9B,4DAImD,0DAJnD,EAGe8B,qBA5DS,UA4DT,qCAhE9B,C;;K;IAwCmD, wH;MAAA,mB;QAC3C,OAAK,iCAAL,CAAIb,gBAAjB,EAA2B,kBAA3B,C;O;K;IAhCZ,yE;MA2BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAIb,QAAjB,EAA2B,UAA3B,EAAuC,IAAvC,C;;QA0B0B, Q;QA5B9B,4DAImD,sEAJnD,EA4B8B,qBAxBS,UAWBT,qCA5B9B,C;;K;IASJ,gC;MAWK,kBAAD,M;MAAA,k BAAC,qEAAD,4DAA2C,S;K;6CAG/C,yB;MAAA,mG;MAAA,yH;MAAA,6F;QAKW,kC;QAAS,4C;O;MALpB,s EAMQ,Y;QACI,Q;QAAA,sC;UAAiB,U;;QACjB,OAAO,oB;O;MARnB,6G;MAAA,oC;QAKkC,Q;QAA9B,mEA A8B,oEAA9B,C;O;KALJ,C;iFC7HA,a;MAC6C,OAAA,MAAa,YAAW,CAAX,C;K;ICM3B,iC;;MAA6E,Q;MAA A,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;:::;IAC/F,2B;MAAA,iD;MAAuB,oBAAK,IAAL,EA AW,IAAX,C;MAAv B,Y;K;IACA,sC;MAAA,iD;MAAuC,oBAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,oC;MAAA,iD;MAAwC, oBAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAI+B,mC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O; ;K;:::;IACnG,+B;MAAA,mD;MAAuB,sBAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,0C;MAAA,mD;MA AuC,sBAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,wC;MAAA,mD;MAAwC,sBAAK,SAAL,EAAGB,KAAh B,C;MAAxC,Y;K;IAGsC,0C;MAA0D,qBAAU,OAAV,EAAmB,KAAhB,C;;K;;IACHG,sC;MAAA,0D;MAAuB,6B AAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,iD;MAAA,0D;MAAuC,6BAAK,OAAL,EAAC,IAAd,C;MAAvC ,Y;K;IACA,+C;MAAA,0D;MAAwC,6BAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG8C,kD;MAA0D,4BAA iB,OAAjB,EAA0B,KAA1B,C;;K;;IACxG,8C;MAAA,kE;MAAuB,qCAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;I ACA,yD;MAAA,kE;MAAuC,qCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,uD;MAAA,kE;MAAwC,qCAA K,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG2C,+C;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACrG,2 C;MAAA,+D;MAAuB,kCAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,sD;MAAA,+D;MAAuC,kCAAK,OA AL,EAAC,IAAd,C;MAAvC,Y;K;IACA,oD;MAAA,+D;MAAwC,kCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K; IAG+C,4C;8BAAwD,O;;K;;IACvG,+C;MAAA,mE;MAAuB,sCAAK,IAAL,C;MAAvB,Y;K;IAGqD,yD;MAA0D, 4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC/G,qD;MAAA,yE;MAAuB,4CAAK,IAAL,EA AW,IAAX,C;MAAvB, Y;K;IACA,gE;MAAA,yE;MAAuC,4CAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,8D;MAAA,yE;MAAwC,4 CAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAGmD,uD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IA C7G,mD;MAAA,uE;MAAuB,0CAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,8D;MAAA,uE;MAAuC,0CAA K,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,4D;MAAA,uE;MAAwC,0CAAK,SAAL,EAAGB,KAAhB,C;MAAxC ,Y;K;IAI2C,wC;sCAAgE,O;;K;;IAC3G,2C;MAAA,+D;MAAuB,kCAAK,IAAL,C;MAAvB,Y;K;IAI0C,uC;8BAA wD,O;;K;;IACIG,0C;MAAA,8D;MAAuB,iCAAK,IAAL,C;MAAvB,Y;K;IAGwC,qC;8BAAwD,O;;K;;IACHG,wC; MAAA,4D;MAAuB,+BAAK,IAAL,C;MAAvB,Y;K;IAIJ,wC;MACmD,mBAAM,OAAE,KAaf,C;;K;;IAC/ C,oC;MAAA,wD;MAAuB,sBAAK,IAAL,Q;MAAvB,Y;K;IACA,+C;MAAA,wD;MAAgC,2BAAK,OAAL,EAAC,IA Ad,C;MAAhC,Y;K;IACA,+C;MAAA,wD;MAAiD,IAAY,I;MAAzB,2BAAa,SAAR,OAAQ,CAAb,EAAYB,sDA AzB,C;MAApC,Y;K;IAG4C,yC;8BAAwD,O;;K;;IACpG,4C;MAAA,gE;MAAuB,mCAAK,IAAL,C;MAAvB,Y;K; IAIyC,sC;8BAAwD,O;;K;;IACjG,yC;MAAA,6D;MAAuB,gCAAK,IAAL,C;MAAvB,Y;K;IAGkD,sD;MAA0D,4B AAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC5G,kD;MAAA,sE;MAAuB,yCAAK,IAAL,EA AW,IAAX,C;MAAvB,Y; K;IACA,6D;MAAA,sE;MAAuC,yCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,2D;MAAA,sE;MAAwC,yCA AK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG0D,8D;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACpH,

0D;MAAA,8E;MAAuB,iDAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,qE;MAAA,8E;MAAuC,iDAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,mE;MAAA,8E;MAAwC,iDAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;6FCIGJ,yB;MAEI,OAAG,GAAG,CAAC,QAAD,C;K;mFAGV,oB;MAEI,OAAJ,GAAL,GAAE,G;K;6ETVN,a;MAK8C,cAAvC,C;K;6ECHP,Y;MAG+C,S;K;IA6B/C,2B;MAG4D,0BA Ae,WAAf,C;K;IAE5D,mC;MAIwF,0BA Ae,WAAf,C;K;IAExF,mC;MAKwE,0BA Ae,WAAf,C;K;IAGxE,4B;MAI8D,Q;MAH1D,aAAkB,GAAL,O;MACtB,aAAkB,GAAL,O;MACtB,YAAiB,C;MACjB,OAAO,QAAQ,MAAR,IAAkB,QAAQ,MAAjC,C;QAAyC,IAAI,KAAJ,IAAa,IAAI,YAAJ,EAAL,oBAAJ,O;;MACtD,OAAO,G;K;IAIX,wD;MAMuC,Q;MALnC,aAAa,MAAO,OAAM,CAN,EAAS,OAAT,C;MA0BpB,IAzBc,MAyBL,OAAL,KAAkB,SAAtB,C;QAZBsB,MA0BIB,UA1BU,MA0BS,O;;MAzBvB,YAAiB,MAAO,O;MACxB,IAAI,UAAU,KAAAd,C;QACI,gBAAgB,O;QACHB,OAAO,QAAQ,OAaf,C;UAAwB,OAAO,YAAP,EAAO,oBAAP,UAAkB,Y;;;MAE9C,OAAO,M;K;IAGX,gD;MAKOB,UAAmB,M;MAJnC,aAAa,KAAM,Q;MACnB,MAAO,OAAP,IAAiB,UAAW,K;MAc5B,IAbc,KAAL,OAAL,KAAkB,SAAtB,C;QAbqB,MACjB,UAdU,KAcS,O;;MAbvB,YAAiB,KAAM,O;MACP,4B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAY,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MAC9C,OAAO,M;K;IAGX,yD;MAEOB,UAAgB,M;MADhC,YAAY,U;MACI,4B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAY,IAAI,cAAJ,EAAL,sBAAJ,YAAe,O;;MAC3C,OAAO,G;K;oFAGX,oB;MACI,IAAI,IAAK,OAAL,KAAkB,SAAtB,C;QACI,YAAc,IAAK,O;;K;0EAI3B,wB;MAA+D,OAAA,MAAa,QAAO,GAAP,EAAY,OAAZ,C;K;IS/F5E,mC;MAOI,kBAAkB,MAAa,eAAc,SAAd,C;MAC/B,iBAAiB,MAAa,eAAc,IAAd,C;MAC9B,OAAW,gBAAe,UAAhB,GAA+B,SAA/B,GAAyC,CAAC,S;K;0ECUrD,2B;MAKyE,OAAA,MAAa,gBAAe,IAAf,C;K;4EAyBtF,2B;MAKsE,OAAA,MAAa,eAAc,IAAd,C;K;kEAGnF,qB;MACgD,OAAA,MAAa,KAAK,UAAAS,GAAT,EAAC,IAAd,C;K;wEACHC,qB;MAAQ,OAAK,SAAY,a;K;0EACxB,qB;MAAQ,OAAK,SAAY,c;K;IC3D5D,0D;MAGI,OAAO,I;K;ICHX,sC;MAMsD,OAAA,SAAY,UAAAS,WAAW,KAAAX,CAAT,C;K;ItDKIE,uC;Mf2nBW,Q;MAAA,IernBgB,KfqNBZ,IAAS,CAAT,IernBY,KfqNBE,IAAS,wBAA3B,C;QAAA,OAAcS,UernBtB,KfqNBsB,C;;QernBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf4nBW,Q;MAAA,IetnBgB,KfsnBZ,IAAS,CAAT,IetnBY,KfsnBE,IAAS,0BAA3B,C;QAAA,OAAcS,UetnBtB,KfsnBsB,C;;QetnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf6nBW,Q;MAAA,IevnBgB,KfunBZ,IAAS,CAAT,IevnBY,KfunBE,IAAS,0BAA3B,C;QAAA,OAAcS,UevnBtB,KfunBsB,C;;QevnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf8nBW,Q;MAAA,IexnBgB,KfwnBZ,IAAS,CAAT,IexnBY,KfwnBE,IAAS,0BAA3B,C;QAAA,OAAcS,UexnBtB,KfwnBsB,C;;QexnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf+nBW,Q;MAAA,IeznBgB,KfynBZ,IAAS,CAAT,IeznBY,KfynBE,IAAS,0BAA3B,C;QAAA,OAAcS,UeznBtB,KfynBsB,C;;QeznBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MfgoBW,Q;MAAA,Ie1nBgB,Kf0nBZ,IAAS,CAAT,Ie1nBY,Kf0nBE,IAAS,0BAA3B,C;QAAA,OAAcS,Ue1nBtB,Kf0nBsB,C;;Qe1nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MfioBW,Q;MAAA,Ie3nBgB,Kf2nBZ,IAAS,CAAT,Ie3nBY,Kf2nBE,IAAS,0BAA3B,C;QAAA,OAAcS,Ue3nBtB,Kf2nBsB,C;;Qe3nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MfkoBW,Q;MAAA,Ie5nBgB,Kf4nBZ,IAAS,CAAT,Ie5nBY,Kf4nBE,IAAS,0BAA3B,C;QAAA,OAAcS,Ue5nBtB,Kf4nBsB,C;;Qe5nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,wC;MfmoBW,Q;MAAA,Ie7nBgB,Kf6nBZ,IAAS,CAAT,Ie7nBY,Kf6nBE,IAAS,0BAA3B,C;QAAA,OAAcS,Ue7nBtB,Kf6nBsB,C;;Qe7nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,2B;MAII,OAAO,cAAa,SAAb,C;K;oFAGX,yB;MAAA,gD;MAAA,4B;QAKI,OAAcS,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAuC,OAAhC,SAAGC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAqC,OAA9B,SAASB,C;O;KALzC,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAcS,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAuC,OAAhC,SAAGC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAwC,OAAjC,SAAiC,C;O;KAL5C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAyC,OAAIC,SAAkC,C;O;KAL7C,C;IAYW,2C;MAAA,8B;MAAS,uB;K;4FACW,Y;MAAQ,OAAA,gBAAy,O;K;6CAC3C,Y;MAAkC,OAAA,gBfunP/B,YAAQ,C;K;oDetnPX,mB;MAAgD,OAAy,WAAZ,gBAAy,EAAS,OAAT,C;K;iDAC5D,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAy,KAAZ,E;K;mDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,uFAAD,CAAJ,C;QAAGC,OAAO,E;MACvC,OAAmB,UAAZ,gBAAy,EAAY,OAAR,C;K;uDAEvB,mB;MAES,Q;MAAL,IAAI,eAAC,uFAAD,CAAJ,C;QAAGC,OAAO,E;MACvC,OAAmB,cAAZ,gBAAy,EAAY,OAAZ,C;K;;IApB/B,6B;MAII,0C;K;IAqBJ,+C;MAAI,OAAy,kBAAL,SAAK,EAkB,KAAIB,C;K;IAqBhB,0C;MASI,OAA

Y,oBAAL,SAAK,C;K;IAehB,0C;MAYI,OAAY,oBAAL,SAAK,C;K;IAkBhB,2C;MAWI,OAAY,cAAL,SAAK,EA  
Ac,KAAAd,C;K;IAGhB,2C;MAWI,OAAY,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAA  
K,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,  
SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,c  
AAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OA  
AY,cAAL,SAAK,EAAC,KAAAd,C;K;IAwHhB,sC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,sC;MAOI,OAAY,gB  
AAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IA  
GhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OA  
AY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,  
C;K;IAoFhB,sC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,sC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAS  
I,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SA  
AK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;  
MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;wFAsGhB,yB;MAAA,8C;MAA  
A,kf;QAmB0E,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAA  
U,SAAV,EAAgB,WAAhB,EAA6B,iBAA7B,EAAGD,UAAhD,EAA4D,QAA5D,C;QACA,OAAO,W;O;KArBX,C;  
wFAwBA,yB;MAAA,8C;MAAA,kf;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;U  
AAA,WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;Q  
ACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kf;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UA  
AA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAMF,iBAAnF,EAAsG,  
UAAtG,EAaKH,QAaIH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kf;QAmBkE,iC;UA  
AA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QAC/H,UAAU,SAAV,EAAyC,WA  
AzC,EAA+E,iBAA/E,EAaK,G,UAAIG,EAA8G,QAA9G,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8  
C;MAAA,kf;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QACj  
I,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KArB  
X,C;wFAwBA,yB;MAAA,8C;MAAA,kf;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,w  
B;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAMF,iBAAnF,EAAsG,UAAtG,EAaKH,QAaIH  
,C;QACA,OAAO,W;O;KArBX,C;uFAwBA,yB;MAAA,8C;MAAA,kf;QAmBwE,iC;UAAA,oBAAYB,C;QAAG,0  
B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QACrI,UAAU,SAAV,EAA4C,WAA5C,EAaqF,iBAArF,EA  
AwG,UAAxG,EAAoH,QAAPh,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kf;QAmB0E,i  
C;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAA6C  
,WAA7C,EAAuF,iBAAvF,EAA0G,UAA1G,EAAsH,QAAtH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAA  
A,8C;MAAA,kf;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,gB;Q  
ACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;K  
ArBX,C;oFAwBA,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OA  
AY,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;  
MwCl7B9C,eAAiB,I;MxCk7BjB,OwCj7BO,K;K;qFxC07BX,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,O  
AAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,c;MAAT,YAA8B,SAAY,Q;MwC/8BjD,eAAiB,I;MxC+8BjB,OwC9  
8BO,K;K;IxCI9BX,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MwCx9B9C,eAAiB,I;MxCw9BjB,OwCv9B  
O,K;K;IxCO9BX,uC;MD5oCI,IAAI,ECspCI,WAAW,CDtpCf,CAAJ,C;QACI,cCqpCoB,0C;QDppCpB,MAAM,gC  
AAyB,OAAQ,WAAjC,C;;MCqpCV,OAAO,SAAS,SAAT,EAAe,cAAU,OAAV,CAAf,C;K;IAGX,uC;MD1pCI,IA  
AI,ECoqCI,WAAW,CDpqCf,CAAJ,C;QACI,cCmqCoB,0C;QDlqCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCm  
qCV,OAAO,SAAS,SAAT,EAAe,eAAW,OAAX,CAAf,C;K;IAGX,uC;MDxqCI,IAAI,ECKrCI,WAAW,CDlrCf,CA  
AJ,C;QACI,cCirCoB,0C;QDhrCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCirCV,OAAO,SAAS,SAAT,EAAe,eA  
AS,OAAT,CAAf,C;K;IAGX,uC;MDtrCI,IAAI,ECgsCI,WAAW,CDhsCf,CAAJ,C;QACI,cC+rCoB,0C;QD9rCpB,  
MAAM,gCAAYB,OAAQ,WAAjC,C;;MC+rCH,WAAS,W;MAAT,YAAsB,gBAAGB,SAAhB,EAAsB,OAAtB,K;M  
wChhC7B,eAAiB,I;MxCghCjB,OwC/gCO,K;K;IxChkCX,uC;MDpsCI,IAAI,EC8sCI,WAAW,CD9sCf,CAAJ,C;Q  
ACI,cC6sCoB,0C;QD5sCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC6sCV,OAAO,SAAS,SAAT,EAAe,iBAAW,  
OAAX,CAAf,C;K;IAGX,uC;MDltCI,IAAI,EC4tCI,WAAW,CD5tCf,CAAJ,C;QACI,cC2tCoB,0C;QD1tCpB,MAA



M,gCAAYB,OAAQ,WAAjC,C;;MC2tCV,OAAO,SAAS,SAAT,EAAe,iBAAY,OAAZ,CAAf,C;K;IAGX,uC;MDhu  
CI,IAAI,EC0uCI,WAAW,CD1uCf,CAAJ,C;QACI,cCyuCoB,0C;QDxuCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;  
MCyuCH,WAAS,c;MAAT,YAAyB,gBAAgB,SAAhB,EAAsB,OAAtB,EAA+B,KAA/B,C;MwC1jChC,eAAiB,I;M  
xC0jCjB,OwCzjCO,K;K;IxC4jCX,uC;MD9uCI,IAAI,ECwvCI,WAAW,CDxvCf,CAAJ,C;QACI,cCuvCoB,0C;QDt  
vCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCuvCH,WAAS,W;MAAT,YAAsB,SAAS,SAAT,EAAe,iBAAU,OA  
AV,CAAf,C;MwCxc7B,eAAiB,I;MxCwkCjB,OwCvkCO,K;K;IxC0kCX,uC;MD5vCI,IAAI,ECuwCI,WAAW,C  
DvwCf,CAAJ,C;QACI,cCswCoB,0C;QDrwCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCswCV,OAAO,gBAAgB  
,SAAhB,EAAsB,OAAtB,EAA+B,IAA/B,C;K;IAGX,sD;MAWI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAs  
C,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAl  
B,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MA  
UI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,O  
AAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAy,SAAY  
,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAt  
C,C;MACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MwC9pC9C,eAAiB,I;MxC8pCj  
B,OwC7pCO,K;K;IxCgqCX,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OA  
AY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EA  
sC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,uD;MAUI,oCAAA,2BAAkB,SA  
AlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACN,WAAS,c;MAAT,YAA8B,SAAY,OAAM,SAAN,EAAiB,OAAjB  
,C;MwCxsCjD,eAAiB,I;MxCwsCjB,OwCvsCO,K;K;IxC0sCX,uD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7  
B,EAAsC,gBAAtC,C;MACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MwCttC9C,eA  
AiB,I;MxCstCjB,OwCrtCO,K;K;IxCwtCX,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC  
/E,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EA  
AyB,OAAzB,C;K;IAGrB,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BA  
AkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;  
K;IAGrB,wD;MAWkD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjF,oCAAA,2BAAkB,SAAlB,EA  
A6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;K;IAGrB,wD;M  
AW8C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC7E,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EA  
AsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;K;IAGrB,wD;MAWgD,yB;QAA  
A,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;  
MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;K;IAGrB,wD;MAWkD,yB;QAAA,YAAiB,C;MA  
AG,uB;QAAA,UAAe,gB;MACjF,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,M  
AAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;K;IAGrB,wD;MAWoD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,U  
AAe,gB;MACnF,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EA  
Ac,SAAd,EAAYB,OAAzB,C;K;IAGrB,yD;MAWsD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACrF,o  
CAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,  
OAAzB,C;K;IAGrB,yD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACR,SAAY,MAAK,OAAL,EAAC,SAAd,EAAYB,OAAzB,C;K;iFA  
GrB,8B;MAKI,OAAy,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;iFAG5B,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwI  
O,gCaxIK,eAAY,OAAZ,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,gBA  
Aa,OAAb,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,gBAAW,OAAX,EA  
wIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,mBAAY,OAAZ,CAwIL,C;O;KA7  
IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,kBAaA,OAAb,EAwIL,C;O;KA7IX,C;iFAQA,y  
B;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,kBAAc,OAAd,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;M  
AxIA,qC;QAKI,OAwIO,gCaxIK,sBA Ae,OA Af,CAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,  
OAwIO,gCaxIK,mBAAY,OAAZ,CAwIL,C;O;KA7IX,C;IAQA,sC;MAKI,OAAO,oBAAoB,SAAPB,EAA0B,QAA  
1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAhB,IAAP,CAAxB,EAAsD,  
SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KA  
AhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK

,EAAO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OA  
AO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBAAO,QA  
AS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL  
,SAAK,EAAO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;M  
AII,OAAO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBA  
AO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;iFAGX,+B;MAKI,OAAy,SAAY,  
QAAO,QAAP,C;K;iFAG5B,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KAL  
X,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;  
MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MA  
AA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,O  
AAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,  
SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,  
QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;K  
ALX,C;8FAQA,8B;MAKI,OAAy,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;IAoBL,2B;MAAsB,OAAA,CAAE,iB  
AAU,CAAV,C;K;IAP/C,2B;MAOI,IAAI,mBAAO,CAAX,C;QAwQY,eAxQO,WAwQP,C;;K;IANhB,2B;MAQI,I  
AAI,mBAAO,CAAX,C;QAAC,UAAU,SAAV,C;K;IAGIB,wC;MAQI,IAAI,mBAAO,CAAX,C;QAAC,cAAc,SAAd  
,EAAoB,UAApB,C;K;IAGIB,gD;MAewD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACvF,oCAAa,2B  
AAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EAA  
wC,cAAxC,C;K;IAGJ,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAa,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KA  
AT,QAAS,C;K;IAGb,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,oCAAa,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KA  
AT,QAAS,C;K;IAGb,gD;MAagC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/D,oCAAa,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KA  
AT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAa,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAA8C,SAA9C,EAAyD,OAAzD,EAAkE,cAAI  
E,C;K;IAGJ,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,oCAAa,2BAAkB,SAAlB,EA  
A6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KAAT,QAAS,  
C;K;IAGb,gD;MAamC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACIE,oCAAa,2BAAkB,SAAlB,EA  
A6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KAAT,QAAS,  
C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAa,2BAAkB,SAAlB,EA  
A6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAPB,C;MACvB,KAAT,QAAS,C  
;K;iFAGb,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,  
SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UA  
AL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;IAGhB,yC;M  
AMI,IAAI,mBAAO,CAAX,C;QAAC,gBAAc,SAAd,EAAoB,UAApB,C;K;IAGIB,+D;MAa0E,yB;QAAA,YAAiB,  
C;MAAG,uB;QAAA,UAAe,gB;MACzG,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gB  
AAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EAAwC,UAAxC,C;K;IAGJ,mC;MAII,OAAO,EAAS,MAAM,MAA  
K,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAA  
M,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EA  
AS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,O  
AAO,EAAS,MAAM,MAAK,SAAL,C;K;IAOH,kD;MAAA,wB;QAAW,qCAAK,KAAL,E;O;K;IAJIC,oC;MAII,O  
AAO,iBAAM,gBAAN,EAAY,gCAAZ,C;K;IuDnpEX,oB;MAAA,wB;MAEI,6B;MACA,gC;MAKuB,UAAT,MAA  
S,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;  
MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CzC2BuB,IyC3BIC,IAA+B,C;;MAInC,qBAAqB  
,48C;MACrB,WAAW,mBAAmB,cAAAnB,EAAmC,UAAAnC,EAA+C,IAA/C,C;MACX,YAAY,eAAS,IAAK,OAAL  
,GAAY,CAAZ,IAAT,C;MACZ,0BAAU,IAAV,e;QACI,MAAM,MAAI,CAAJ,IAAN,IAAe,MAAM,GAAN,IAAW,  
KAAK,GAAL,CAAX,I;;MAEnB,yBAAoB,K;MAGpB,oBAAoB,m/D;MACpB,4BAAuB,mBAAmB,aAAAnB,EAA

kC,UAAIC,EAA8C,IAA9C,C;K;;IAvB/B,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IA2BA,qC;MAKkB,IAJP,I;MA  
CH,WAAO,EAAP,C;QA Ae,W;WACf,WAAO,IAAP,C;QAAgB,OAAI,CAAC,KAAO,CAAR,MAAc,CAAlB,GAA  
qB,QAAS,CAA9B,GAAqC,OAAS,E;;QAEID,QAAM,KAAK,CAAX,C;eACI,C;YA AK,eAAS,E;YAA d,K;eACA,  
C;YA AK,OAA C,QAAS,CAAV,GAAiB,E;YAAtB,K;;YACQ,cAAS,E;YAHrB,K;;MAJR,W;K;IAYJ,qC;MAII,SA  
AS,SzCRiC,I;MyCU1C,YAA Y,kBAaKB,sBAAS,kBAA3B,EAA8C,EAA9C,C;MACZ,YAA Y,sBAAS,kBAAT,CA  
A2B,KAA3B,C;MACZ,WAAW,sBAAS,qBAAT,CAA8B,KAA9B,C;MACX,YAA Y,kBAaKB,IAAlB,EAAwB,KA  
AK,KAAL,IAAxB,C;MAEZ,OAAW,UAAS,EAAb,GAAyC,mDAAzC,GAAoD,K;K;IAG/D,8D;MAKiB,UAIE,M;  
MARf,aAAa,eAAS,YAAT,C;MACb,YAA Y,C;MACZ,UAAU,C;MACV,YAA Y,C;MACC,yB;MAAb,OAAa,cAAb  
,C;QAAa,iC;QACT,aAAa,WAAW,IzCxBc,IyCwBzB,C;QACb,MAAM,MAAQ,CAAC,SAAW,EAAZ,KAA sB,K;  
QACpC,IAAI,SAAS,EAAb,C;UACI,OAAO,cAAP,EAAO,sBAAP,YAAkB,G;UACIB,MAAM,C;UACN,QAAQ,C;  
;UAER,gBAAS,CAAT,I;;MAGR,OAAO,M;K;ICIE X,+B;MAII,eAAe,CAAC,iBAAO,CAAP,IAAD,IAAa,CAAb,I  
;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,2B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QAC  
I,UAAU,sBAAK,KAAL,C;QACV,sBAAK,KAAL,EAAc,sBAAK,YAAL,CAAd,C;QACA,sBAAK,YAAL,EAAqB  
,GAAR,B,C;QACA,mC;;K;IrDbR,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAM,B,C;K;mFAG9  
B,yB;MAkBA,iB;MAIBA,uB;QAMI,OakBO,MAAO,KAIBC,CakBD,EAIBY,CakBZ,C;O;KAXBIB,C;mFASA,y  
B;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O;KAFiB,C;mFASA,yB;MAAA,iB;  
MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,gB;MAMI,OAAW,kBAAK  
,CAAL,MAAJ,GAAY,CAAZ,GAAM,B,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CA  
AJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,C  
AAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mF  
AGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KAN  
IB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;  
O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,  
C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAoEe,CAPeF,GAoEkB,C;MAAzB,  
OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAM,B,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,  
OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QA  
QI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU,  
C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EA AW,CAAX,C;;MACV  
B,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAi  
B,MAxHV,MAAO,KAwHe,GAXHf,EAwHoB,CAXHpB,C;;MAYhd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UA  
AU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAAO,KakIe,GAlIf,EAkIoB,CAlIp  
B,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAA  
V,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,  
UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7  
IvB,MAAJ,GAAY,GA AZ,GA6I2B,C;;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAA  
U,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO  
,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,  
MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IAGX,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,G  
AAY,CAAZ,GAAM,B,C;K;mFAG9B,yB;MAkBA,iB;MAIBA,uB;QAMI,OakBO,MAAO,KAIBC,CakBD,EAIBY,  
CakBZ,C;O;KAXBIB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O  
;KAFiB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;m  
FASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAM,B,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB  
;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAA  
O,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAA  
N,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CA  
AjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EA  
AiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CA  
AJ,EAAO,CAAP,EAAU,CAAV,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAo

Ee,CApEf,GAoEkB,C;MAAzB,OAAa,CApEF,iBAAK,GAAL,MAAJ,GAoEM,CApEN,GAAMb,G;K;mFAuE9B,y  
B;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;mFAW  
A,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;IAW  
A,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,G  
AAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;  
QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAxHf,EAwHoB,CxHpB,C;;MAyHd,OAAO,G;K;I  
AGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAiIV,MAA  
O,KAkIe,GAlIf,EAKIoB,CAlIpB,C;;MAMld,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,  
KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G  
;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,  
G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAy,GAAZ,GA6I2B,C;;MACiC,OAAO,G;K;IAGX,4B;MAQc,Q;  
MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,E  
A8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,  
QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IsDvaX,iB;MAA  
A,qB;MAEI,0BAA0B,gBACtB,EADsB,EACd,IADc,EACN,IADM,EACE,IADF,EACU,IADV,EACkB,IADIB,EA  
C0B,IAD1B,EACkC,IADIC,EAC0C,IAD1C,EACkD,IADID,EAC0D,IAD1D,EACkE,IADIE,EAC0E,IAD1E,EACK  
F,IADIF,EAC0F,IAD1F,EACkG,IADIG,EAC0G,IAD1G,EACKH,IADIH,EAC0H,IAD1H,EACKI,IADII,EAETB,IA  
FsB,EAEd,IAFc,EAEN,IAFM,EAEE,IAFF,EAEU,IAFV,EAEB,IAFIB,EAEOB,IAF1B,EAekC,IAFIC,EAEOC,IA  
F1C,EAekD,KAFID,EAEOB,KAF1D,EAekE,KAFIE,EAEOE,KAF1E,EAekF,KAFIF,EAEOF,KAF1F,EAekG,KA  
FIG,EAEOG,KAF1G,E;K;;IAF9B,6B;MAAA,4B;QAAA,W;;MAAA,qB;IAQA,0C;MAKI,aAAa,C;MACb,UAA  
U,KAAM,OAAN,GAAa,CAAb,I;MACV,aAAa,E;MACb,YAAy,C;MACZ,OAAO,UAAU,GAAjB,C;QACI,SAAS  
,CAAC,SAAS,GAAT,IAAD,IAAiB,CAAjB,I;QACT,QAAQ,MAAM,MAAN,C;QACR,IAAI,SAAS,KAAb,C;UAC  
I,SAAS,SAAS,CAAT,I;aACR,IAAI,WAAU,KAAd,C;UACD,OAAO,M;;UAEP,MAAM,SAAS,CAAT,I;;MAEd,O  
AAO,UAAc,SAAS,KAAb,GAAoB,CAApB,GAA2B,CAArC,K;K;IAGX,mC;MAKI,SAAS,S3CCiC,I;M2CA1C,Y  
AAy,kBAAkB,mBAAM,mBAAxB,EAAoC,EAAPC,C;MACZ,WAAW,KAAK,mBAAM,mBAAN,CAAiB,KAAj  
B,CAAL,I;MACX,OAAW,OAAO,EAAX,GAAe,IAAf,GAAyB,E;K;IAGpC,gC;MAIL,OAAO,6BAAoB,C;K;IC7C/  
B,kB;MAAA,sB;MAEI,6B;MACA,8B;MACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,  
kE;MACf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,  
WAAW,oBAAS,CAAT,C5C0BuB,I4C1BIC,IAA+B,C;;MAInC,qBAAqB,sW;MACrB,WAAW,mBAAmB,cAAnB,  
EAAmC,UAAnc,EAA+C,GAA/C,C;MACX,YAAy,eAAS,IAAK,OAAAd,C;MACZ,0BAAU,IAAV,e;QACI,IAAI,  
QAAK,CAAT,C;UAAy,MAAM,GAAN,IAAW,KAAK,GAAL,C;;UACIB,MAAM,GAAN,IAAW,MAAM,MAAI,  
CAAJ,IAAN,IAAe,KAAK,GAAL,CAAf,I;;MAEpB,yBAAoB,K;MAGpB,kBAAkB,0U;MACIB,0BAAqB,mBAA  
mB,WAAAnB,EAAgC,UAAhC,EAA4C,GAA5C,C;MAGrB,oBAAoB,i8B;MACpB,4BAAuB,mBAAmB,aAAnB,E  
AAkC,UAAIC,EAA8C,GAA9C,C;K;;IA7B/B,8B;MAAA,6B;QAAA,Y;;MAAA,sB;K;IAiCA,iC;MAIL,OAAO,6B  
AAmB,C;K;IAG9B,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wB5CRM,a4CQN,C;;MAApC,W;K;IAGJ,oC;M  
AIW,wCAAmB,C;MAAnB,U;QAA6B,wB5CfM,a4CeN,C;;MAApC,W;K;IAGJ,kC;MAQI,SAAS,S5C1BiC,I;M4C  
2B1C,YAAy,kBAAkB,oBAAO,kBAAzB,EAA4C,EAA5C,C;MAEZ,iBAAiB,oBAAO,kBAAP,CAAYB,KAAZB,C  
;MACjB,eAAe,aAAa,oBAAO,mBAAP,CAA0B,KAA1B,CAAb,GAAGD,CAAhD,I;MACf,WAAW,oBAAO,qBAA  
P,CAA4B,KAA5B,C;MAEX,IAAI,KAAK,QAAT,C;QACI,OAAO,C;;MAGX,kBAAkB,OAAS,C;MAE3B,IAAI,g  
BAAE,CAAnB,C;QACI,YAAy,C;QACZ,gBAAGB,U;QACHb,aAAU,CAAV,OAAa,CAAb,M;UACI,yBAAc,QAA  
S,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAy,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;UACA,yB  
AAc,QAAS,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAy,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;;  
QAEJ,OAAO,C;;MAGX,IAAI,QAAQ,CAAZ,C;QACI,OAAO,W;;MAGX,eAAgB,KAAK,UAAL,I;MACHb,cAAg  
B,QAAQ,EAAZ,GAAB,WAAW,CAA7B,GAAoC,Q;MACHd,OAAQ,SAAU,IAAI,OA AJ,IAAV,CAAD,GAA2B,  
C;K;ICnGtC,0B;MAAA,8B;MACI,+BAA+B,gBAC3B,GAD2B,EACnB,GADmB,EACX,GADW,EACH,GADG,E  
ACK,GADL,EACa,GADb,EACqB,GADrB,EAC6B,IAD7B,EACqC,IADrC,EAC6C,IAD7C,EACqD,IADrD,EAC6  
D,IAD7D,EACqE,IADrE,EAC6E,IAD7E,EACqF,IADrF,EAC6F,KAD7F,EACqG,KADrG,EAC6G,KAD7G,EACq  
H,KADrH,EAC6H,KAD7H,E;MAG/B,gCAAAGC,gBAC5B,CAD4B,EACzB,CADyB,EACtB,CADsB,EACnB,CAD

mB,EACHB,CADgB,EACb,CADa,EACV,CADU,EACP,EADO,EACH,CADG,EACA,EADA,EACI,CADJ,EACO,  
CADP,EACU,EADV,EACc,EADd,EACkB,EADIB,EACsB,CADtB,EACyB,CADzB,EAC4B,CAD5B,EAC+B,CA  
D/B,EACKc,CADIC,E;K;;IAJpC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;IASA,qC;MACI,YAAy,kBAAkB,4BA  
Ae,wBAAjC,EAakD,SAaID,C;MACZ,OAAO,SAAS,CAAT,IAAc,aAAO,4BA Ae,wBAAf,CAA+B,KAA/B,IAAw  
C,4BA Ae,yBAAf,CAAgC,KAAhC,CAAxC,IAAP,C;K;ICXzB,qC;MACI,OAAe,IAAR,8BAAgB,IAAhB,KACY,I  
AAR,8BAAgB,IADpB,C;K;ICCX,wC;M5CiBW,Q;MAAA,I4CXgB,K5CWZ,IAAS,CAAT,I4CXY,K5CWE,IAAS,  
2BAA3B,C;QAAA,OAAc,qB4CXtB,K5CWsB,C;;Q4CXb,MAAM,8BAA0B,mCAAyB,gBAAzB,MAA1B,C;;M  
AAtC,W;K;ICRJ,sC;MAEI,WAAW,ShDkC+B,I;MgDhC1C,IAAY,GAAR,oBAAgB,GAAhB,KAAkC,GAAR,oBA  
AgB,GAA1C,CAAJ,C;QACI,OAA8B,OAAtB,KAAK,CAAC,OAAO,CAAP,IAAD,IAAa,CAAb,IAAL,KAAcB,C;;  
MAGIC,IAAY,IAAR,oBAAgB,IAAhB,KAAkC,IAAR,oBAAgB,IAA1C,CAAJ,C;QACI,OAAO,S;;MAEX,OAAO,  
wB;K;ICPX,wC;MxCqTe,WwC7SY,KxC6SZ,IAAS,C;MAAT,S;QAAc,OwC7SF,KxC6SE,IAqgHT,gBAAR,iBAA  
Q,C;;MArgHT,U;MAAA,S;QAAA,SAAsC,sBwC7StB,KxC6SsB,C;;QwC7Sb,MAAM,8BAA0B,iCAAuB,cAAvB,  
MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCsTe,WwC9SY,KxC8SZ,IAAS,C;MAAT,S;QAAc,OwC9SF,KxC8SE,IAig  
HT,gBAAR,iBAAQ,C;;MAjgHT,U;MAAA,S;QAAA,SAAsC,sBwC9StB,KxC8SsB,C;;QwC9Sb,MAAM,8BAA0B  
,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCuTe,WwC/SY,KxC+SZ,IAAS,C;MAAT,S;QAAc,OwC/  
SF,KxC+SE,IA6/GT,gBAAR,iBAAQ,C;;MA7/GT,U;MAAA,S;QAAA,SAAsC,sBwC/StB,KxC+SsB,C;;QwC/Sb,  
MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCwTe,WwChTY,KxCgTZ,IAAS,C;MA  
AT,S;QAAc,OwChTF,KxCgTE,IAy/GT,gBAAR,iBAAQ,C;;MAz/GT,U;MAAA,S;QAAA,SAAsC,sBwChTtB,Kx  
CgTsB,C;;QwChTb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IASO,6C;MAAA,8B;MAAS,uB;  
K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAgD,OAA  
A,gBAAY,gBAAS,OAAT,C;K;mDAC5D,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SAAzB,C;MACb,OAAO,6  
BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OxCs  
rBO,UwCtrBA,gBxCsrBR,QAAQ,EwCtrBoB,O3EgOF,KmCsdIB,C;K;yDwCprBX,mB;MAES,Q;MAAL,IAAI,eA  
AC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OxCy6BO,cwCz6BA,gBxCy6BR,QAAQ,EwCz6BwB,O3E2NN,  
KmC8sBIB,C;K;;IwC/7BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBA  
AY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;m  
DAC7D,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;  
MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OxCqqBO,UwCrqBA,gBxCqqBR,QA  
AQ,EwCrqBoB,O3DgNA,KmBqdpB,C;K;yDwCnqBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAi  
C,OAAO,E;MACxC,OxCw5BO,cwCx5BA,gBxCw5BR,QAAQ,EwCx5BwB,O3D2MJ,KmB6sBpB,C;K;;IwC96Bn  
B,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MA  
AkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAa  
,2BAAkB,KAAIB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eA  
AC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OxCopBO,UwCppBA,gBxCopBR,QAAQ,EwCppBoB,O5EkIA,K  
oCkhBpB,C;K;yDwClpBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OxCu4  
BO,cwCv4BA,gBxCu4BR,QAAQ,EwCv4BwB,O5E6HJ,KoC0wBpB,C;K;;IwC75BnB,8B;MAMI,4C;K;IA2BO,6C  
;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDA  
C9C,mB;MAAkD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC9D,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SA  
AzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,  
OAAO,E;MACzC,OxCmoBO,UwCnoBA,gBxCmoBR,QAAQ,EwCnoBoB,O1EkHE,KkCihBtB,C;K;yDwCjoBX,  
mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OxCs3BO,cwCt3BA,gBxCs3BR,Q  
AAQ,EwCt3BwB,O1E6GF,KkCywBtB,C;K;;IwC54BnB,8B;MAMI,4C;K;ICtIJ,qC;MAII,SAAS,SID+BiC,I;MkD9  
B1C,OAAa,CAAN,gBAAc,EAAd,KACU,EAAN,gBAAc,EADIB,KA EI,OAAM,GAFV,KAGI,KAAK,IAAL,KAC  
C,OAAM,IAAN,KACS,IAAN,gBAAc,IADjB,KAEG,OAAM,IAFT,IAGG,OAAM,IAHT,IAIG,OAAM,IAJT,IAK  
G,OAAM,IALT,IAMG,OAAM,KAPV,CAHJ,C;K;;mCCTP,gB;;K;;ICAJ,wB;K;;IAIA,wB;K;;IAIA,wB;K;;IAKiC  
,uB;MAAC,oB;QAAA,OAA0B,E;MAA1B,gB;K;;IAEiC,kB;K;;IAqCqC,sB;MAAC,gB;K;;IAgCN,4B;MAAC,sB;  
K;;IAEjC,uB;K;;IA8DmC,4B;MAAC,kB;K;;IAEpC,oB;K;;IAmCA,+B;K;;ICvLA,oB;K;;IAIA,wB;K;;oF7DLA,qB;  
MAKqE,uCoCHtB,E;K;IgpCK/C,yB;MAAA,kD;MAAA,4B;QAQsE,mBAAY,SAAZ,C;O;KARtE,C;IAUA,iC;MA

GI,OAA sB,UAA Y,QA A vB,KAA mC,SAA 9C,GAC e,UAA Y,UAD 3B,GAG I,gBAA gB,UAA hB,C;K;IAGR,qC;MA EI,YoC1B2C,E;MpC2B3C,eAA e,UAA W,W;MAC1B,OAA O,QAAS,UAA hB,C;QACU,KAA Y,MAAK,QAAS,OA Ad,C;MAC tB,OAA O,K;K;IAGX,8C;MAQc,Q;MANV,IAAI,KAAM,OAAN,GAA a,UAA W,KAA 5B,C;QACI,OA AO,gBAA gB,UAA hB,C;;MAEX,eAA e,UAA W,W;MAC1B,YAA Y,C;MACZ,OAA O,QAAS,UAA hB,C;QACI,MA AM,YAAN,EAAM,oBAAN,UAA iB,QAAS,O;;MAE9B,IAAI,QAA Q,KAAM,OAA iB,C;QACI,MAAM,KAAN,IA Ae,I;;MAEnB,OAA O,K;K;IAIX,yB;MAG6C,sBAAY,OAA Z,E;K;wGAE7C,yB;MAAA,+D;MAAA,gC;QAI0B,gB AAF,gB;QAA qB,aJW5B,W;QIXA,OJYO,SIZoC,Q;O;KAJ/C,C;yGAOA,yB;MAAA,4E;MAAA,gE;MAAA,0C;QA II,qBAA qB,QAA rB,C;QAC8B,gBAA vB,eAA a,QAA b,C;QAA6B,aJGpC,W;QIHA,OJIO,SIJ4C,Q;O;KALvD,C;IA SA,wB;MAG2C,oBAAU,OAA V,E;K;sGAE3C,yB;MAAA,uE;MAAA,gC;QAI8B,gBAA nB,oB;QAA yB,aJVhC,W;QIU A,OJTO,SISwC,Q;O;KAJnD,C;wGAOA,yB;MAAA,wE;MAAA,0C;QAI sC,gBAA 3B,mBAA iB,QAA jB,C;QA AiC,aJjBxC,W;QIiBA,OJhBO,SIgBgD,Q;O;KAJ3D,C;IAQA,qB;MAIuD,oBAAU,IAAV,E;K;sGAEvD,yB;MAAA ,wE;MAAA,gC;QAIiC,gBAA tB,oB;QAA4B,aJ/BnC,W;QI+BA,OJ9BO,SI8B2C,Q;O;KAJtD,C;uGAOA,yB;MAA A,uE;MAAA,0C;QAIyC,gBAA 9B,mBAA oB,QAA pB,C;QAA oC,aJtC3C,W;QIsCA,OJrCO,SIqCmD,Q;O;KAJ9D, C;IAQA,mC;MAOqB,Q;MAAA,kC;MAAjB,iBAAc,CAAd,yB;QACI,sBAAK,KAAL,EAAc,KAAd,C;;K;IAIR,+B; MAMuD,sBAAQ,4BAAR,C;K;IAEvD,6B;MAIwE,kBAAhB,0B;MAAwB,uB;MAAxB,OJjE7C,W;K;IImEX,4B;M AQI,gBAAgB,SAAhB,EAA sB,cAA tB,C;K;IAGJ,2C;MAQI,gBAAgB,SAAhB,EAA sB,UAA tB,C;K;IAGJ,2C;MA CI,IAAI,IAAK,KAAL,IAAa,CAA jB,C;QAA oB,M;MAEpB,YAA Y,YAA Y,IAAZ,C;MACZ,gBAAc,KAAd,EAAq B,UAA rB,C;MAEA,aAAU,CAAV,MAAkB,KAAM,OAAxB,M;QACI,iBAAK,CAAL,EAAU,MAAM,CAAN,CA AV,C;;K;IAIR,uC;MACI,OAAO,gBAAkB,IAAI,O;K;IAGX,iF;MAII,oCAAa,2BAAkB,UAAIB,EAA8B,QAA9B ,EAAwC,MAAO,OAA/C,C;MACb,gBAAgB,WAAW,UAA X,I;MACHb,oCAAa,2BAAkB,iBAAlB,EAAqC,oBAA oB,SAApB,IAArC,EAAoE,WAA Y,OAAhF,C;MAEb,IAAI,WAAkB,QAAO,WAAP,CAAIB,IAAYC,WAAkB,QA AO,MAAP,CAA/D,C;QACI,eAA sB,MAAY,UAA S,UAA T,EAAqB,QAA rB,C;QACtB,WAA Y,KAAL,QAAJ,EAA c,iBAAd,C;;QAExB,IAAI,WAAW,WAA X,IAA0B,qBAAqB,UAA nD,C;UACI,iBAAc,CAAd,UAA sB,SAAtB,U;Y ACI,YAA Y,oBAA oB,KAAPB,IAAZ,IAAYC,OAAO,aAAa,KAAb,IAAP,C;;UAG7C,mBAAc,YAA Y,CAAZ,IAA d,aAAmC,CAAnC,Y;YACI,YAA Y,oBAA oB,OAAPB,IAAZ,IAAYC,OAAO,aAAa,OAAb,IAAP,C;;K;8GAMzD, qB;MAEgF,gB;K;kGAehF,yB;MAAA,4D;MAAA,4B;QAC8E,OAAK,aAAL,SAAK,C;O;KADnF,C;sGAIA,gC;M AEI,OAAI,SAAJ,GAEL,SAFJ,GAII,SN83BoB,Q;K;IM13B5B,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ, OAAO,K;K;IAGX,mC;MAEL,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ,OAAO,K;K;IAIX,mC;MAIqD,mB;K;IAEr D,wC;MPzNI,IAAI,EOgOI,YAA Y,CPhOhB,CAAJ,C;QACI,cO+NqB,gC;QP9NrB,MAAM,gCAAyB,OAAQ,WA AjC,C;;K;IOiOd,8C;MAAoE,Y;K;I8D1PV,qC;MAAiC,6B;K;uDAIvF,mB;MACI,qB;MACA,eAAe,e;MACf,OAA O,QAAS,UAAhB,C;QACI,IAAI,OAAA,QAAS,OAAT,EAAMB,OAAnB,CAAJ,C;UACI,QAAS,S;UACT,OAAO,I ;;MAGf,OAAO,K;K;yDAGX,oB;MAGoB,Q;MAFhB,qB;MACA,eAAe,K;MACC,0B;MAAhB,OAAgB,cAAhB,C ;QAAgB,yB;QACZ,IAAI,eAAI,OAAJ,CAAJ,C;UAAkB,WAAW,I;;MAEjC,OAAO,Q;K;IAKuC,sE;MAAA,qB;Q AAE,OAA M,gBAAN,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC,YAA5B,iEAA4B,EAAU,oDA AV,C;K;IAKU,sE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC ,YAA5B,iEAA4B,EAAU,oDAAV,C;K;gDAGxC,Y;MACI,qB;MACA,eAAe,IAAK,W;MACpB,OAAO,QAAS,UA AhB,C;QACI,QAAS,O;QACT,QAAS,S;;K;iDAIjB,Y;MAE8B,OAAA,IAAK,U;K;yDAGnC,Y;K;;IC3CgD,+B;MA AiC,oC;MACjF,gBAA8B,C;K;8CAM9B,mB;MAMI,qB;MACA,iBAAI,SAAJ,EAAU,OAAV,C;MACA,OAAO,I; K;mDAGX,2B;MAMc,UACF,M;MANR,oCAAa,4BAAMB,KAANB,EAA0B,SAAI1B,C;MAEb,qB;MACA,aAAa, K;MACb,cAAc,K;MACJ,0B;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,kBAAL,eAAJ,EAAI,uBAAJ,WAAc,CA Ad,C;QACA,UAAU,I;;MAEd,OAAO,O;K;0CAGX,Y;MACI,qB;MACA,yBAA Y,CAAZ,EAAe,SAAf,C;K;IAKiB, gE;MAAA,qB;QAAE,OAA M,gBAAN,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;IAK U,gE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;6C AIX,Y;MAAqD,iD;K;mDAErD,mB;MAAoD,0BAAQ,OAA R,KAA oB,C;K;kDAExE,mB;MACqB,Q;MAAA,6B; MAAjB,iBAAc,CAAd,yB;QACI,IAAI,wBAAI,KA AJ,GAAC,OAAd,CAAJ,C;UACI,OAAO,K;;MAGf,OAAO,E;K ;sDAGX,mB;MACI,iBAAc,sBAAd,WAA+B,CAA/B,U;QACI,IAAI,wBAAI,KA AJ,GAAC,OAAd,CAAJ,C;UACI, OAAO,K;;MAGf,OAAO,E;K;iDAGX,Y;MAA6D,iCAAa,CAAb,C;K;yDAC7D,iB;MAAuE,sDAAiB,KA AjB,C;K ;oDAGvE,8B;MAA4E,uCAAQ,IAAR,EAAc,SAAd,EAAyB,OAAzB,C;K;wDAE5E,8B;MAII,eAAe,0BAAa,SAAb

,C;MACf,YAAO,UAAU,SAAV,I;MnEuDX,iBAAc,CAAd,UAAsB,KAAtB,U;QmEtDiB,e;QACA,iB;;K;2CAIjB,i  
B;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,oC  
AAa,uBAAc,IAAd,EAAoB,KAApB,C;K;6CAGxB,Y;MAG+B,OAAA,oCAAa,yBAAgB,IAAhB,C;K;IAG5C,kD;  
MAAA,oB;MACI,eACsB,C;MACtB,cAIqB,E;K;yDAErB,Y;MAAkC,sBAAQ,gB;K;sDAE1C,Y;MAEW,Q;MADP  
,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,eAAO,mBAAP,EAAO,2BAAP,O;MACA,OAAO,wBAAI,WA  
AJ,C;K;wDAGX,Y;MtE5CJ,IAAI,EsE6CU,gBAAQ,EtE7CIB,CAAJ,C;QACI,csE4CwB,sE;QtE3CxB,MAAM,6BA  
AsB,OAAQ,WAA9B,C;;MsE6CF,6BAAS,WAAT,C;MACA,eAAQ,W;MACR,cAAO,E;K;;IAOqB,6D;MAHpC,o  
B;MAGmD,wD;MAG3C,oCAAa,4BAAmB,KAAmB,EAA0B,WAAyB,KAAmD,C;MACb,eAAa,K;K;iEAGjB,Y;M  
AAsC,sBAAQ,C;K;+DAE9C,Y;MAAgC,mB;K;8DAEHc,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B;M  
AE1B,eAAO,mCAAP,EAAO,YAAP,C;MACA,OAAO,wBAAI,WAAJ,C;K;mEAGX,Y;MAAoC,sBAAQ,CAAR,I;  
K;+DAEpC,mB;MACI,wBAAI,YAAJ,EAAW,OAAX,C;MACA,mC;MACA,cAAO,E;K;+DAGX,mB;MtEIFJ,IAA  
I,EsEmFU,gBAAQ,EtEnFIB,CAAJ,C;QACI,csEkFwB,4E;QtEjFxB,MAAM,6BAAsB,OAAQ,WAA9B,C;;MsEkFF  
,wBAAI,WAAJ,EAAU,OAAV,C;K;;IAIgb,+D;MAAuF,8B;MAAtF,kB;MAA0C,4B;MAC/D,eAAyB,C;MAGrB,o  
CAAa,2BAAkB,gBAAlB,EAA6B,OAA7B,EAAc,WAAK,KAA3C,C;MACb,eAAa,UAAU,gBAAV,I;K;wDAGjB  
,0B;MACI,oCAAa,4BAAmB,KAAmB,EAA0B,YAA1B,C;MAEb,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OA  
AvB,C;MACL,mC;K;wDAGJ,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,OAAO,wBAAK,mB  
AAY,KAAZ,IAAL,C;K;6DAGX,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,aAAa,WAAK,kB  
AAS,mBAAY,KAAZ,IAAT,C;MACIB,mC;MACA,OAAO,M;K;wDAGX,0B;MACI,oCAAa,2BAAkB,KAAIB,EA  
AyB,YAAzB,C;MAEb,OAAO,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OAAvB,C;K;mGAGO,Y;MAAQ,mB  
;K;2DAE/B,Y;MAA+C,WAAK,iB;K;;ICxMN,8B;MAAiC,sB;MAwCnF,uBAAoC,I;MA+CpC,yBAA6C,I;K;IAIF  
R,oD;MAAC,wB;MAGIC,gBAAqB,K;K;iFAHa,Y;MAAA,yB;K;uGAKZ,Y;MAAQ,oB;K;8DAE9B,oB;MAKI,eA  
Ae,IAAK,S;MACpB,gBAAc,Q;MACd,OAAO,Q;K;wDAGX,Y;MAA+B,iEAAc,IAAd,C;K;wDAC/B,Y;MAAKC,i  
EAAc,IAAd,C;K;sDACIC,iB;MAA4C,+DAAY,IAAZ,EAakB,KAAIB,C;K;;IAIB5C,8E;MAAA,wE;MAAsC,2CA  
AK,KAAAM,IAAX,EAAGB,KAAAM,MAAtB,C;MAAtC,Y;K;IASBJ,+C;MACsE,6B;K;mEACIE,mB;MAAmD,kCA  
Ac,OAAd,C;K;iEAEnD,mB;MAAiD,gCAAY,OAAZ,C;K;;yCAIrD,Y;MACI,YAAQ,Q;K;IAOQ,+F;MAAA,sD;M  
AAS,6B;K;uFACb,mB;MAAwC,MAAM,qCAA8B,8BAA9B,C;K;mFAC9C,Y;MACI,4BAAwB,Q;K;4FAG5B,mB  
;MAAsD,sDAAY,OAAZ,C;K;IAI3C,oH;MAAA,kD;K;4GACH,Y;MAAkC,OAAA,0BAAc,U;K;yGACHd,Y;MAA  
yB,OAAA,0BAAc,OAAO,I;K;2GAC9C,Y;MAAwB,0BAAc,S;K;;sFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5  
B,6G;K;0FAOJ,mB;MACI,qB;MACA,IAAI,+CAAY,OAAZ,CAAJ,C;QACI,4BAAwB,cAAO,OAAP,C;QACxB,O  
AAO,I;;MAEX,OAAO,K;K;oIAGY,Y;MAAQ,OAAA,4BAAwB,K;K;4FAEvD,Y;MAAsC,4BAAwB,iB;K;;0FA9B  
1E,Y;MACI,IAAI,4BAAJ,C;QACI,6F;;MA+BJ,OAAO,mC;K;kDAKf,gB;MAEyB,Q;MADrB,qB;MACqB,OAAA,  
I9E8Q2D,QAAQ,W;M8E9QxF,OAAqB,cAArB,C;QAAqB,wB;QAAf,U9EiMsD,U;Q8EjMjD,Y9E8MiD,Y;Q8E7  
MxD,iBAAI,GAJ,EAAS,KAAT,C;;K;IAQc,iG;MAAA,sD;MAAS,oC;K;yFACf,mB;MAAwC,MAAM,qCAA8B,  
gCAA9B,C;K;qFAC9C,Y;MAAuB,4BAAwB,Q;K;8FAE/C,mB;MAAsD,wDAAc,OAAd,C;K;IAI3C,sH;MAAA,k  
D;K;8GACH,Y;MAAKC,OAAA,0BAAc,U;K;2GACHd,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;6GAC9C,Y;MAA  
wB,0BAAc,S;K;;wFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5B,+G;K;sIAOmB,Y;MAAQ,OAAA,4BAAwB,K;  
K;8FAEvD,Y;MAAsC,4BAAwB,iB;K;;4FAnB1E,Y;MACI,IAAI,8BAAJ,C;QACI,iG;;MAoBJ,OAAO,qC;K;gDA  
Gf,e;MACI,qB;MACA,WAAW,YAAQ,W;MACnB,OAAO,IAAK,UAAZ,C;QACI,YAAy,IAAK,O;QACjB,QAA  
Q,KAAAM,I;QACd,IAAI,YAAO,CAAP,CAAJ,C;UACI,YAAy,KAAAM,M;UACIB,IAAK,S;UACL,OAAO,K;;MA  
Gf,OAAO,I;K;kDAIX,Y;K;;IC3I+C,8B;MAAiC,oC;K;0CAEHf,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;  
MAC3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,mCAAY,mBAAU,IAAV,EAAGB,KAAhB,C;K;4CAG  
vB,Y;MAG+B,OAAA,mCAAY,2BAAkB,IAAlB,C;K;;ICbT,0B;MAAuD,8B;MAAIC,4B;MACvD,4BAAkC,K;K;g  
CAkBiC,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;qCAGX,Y;K;iDAGA,uB;K;iFAG8B,Y;MAAQ,OAAA,  
oBAAM,O;K;sCAC5C,iB;MACyC,Q;MAAA,oCAAM,0BAAW,KAAAX,CAAN,4D;K;sCACzC,0B;MAIW,IAAa,I;  
MAHpB,qB;MACA,0BAAW,KAAAX,C;MAEoB,gBAAb,qBAAM,KAAAN,C;MAAqB,qC;MAA5B,OAAO,CAAa,O  
tE8BjB,SsE9BI,2D;K;oCAGX,mB;MACI,qB;MACM,oBAAY,MAAK,OAAL,C;MACIB,qC;MACA,OAAO,I;K;s  
CAGX,0B;MACI,qB;MACM,oBAAY,QAAO,mCAAoB,KAApB,CAAP,EAAmC,CAAnC,EAAc,OAAtC,C;MA  
CIB,qC;K;yCAGJ,oB;MACI,qB;MACA,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE/B,uBAAA,oBxEioDoB,Q

MjrD0C, YkEgDrD, QIEhDqD, CNirD1C, C; MwEhoDpB, qC; MACA, OAAO, I; K; yCAGX, 2B; MACI, qB; MACA, mC  
AAoB, KAAPB, C; MAEA, IAAI, UAAS, SAAb, C; QAAMb, OAAO, oBAAO, QAAP, C; MAC1B, IAAI, QAAS, UAAb, C  
; QAAwB, OAAO, K; MAE3B, IADE, KACF, e; QAAQ, OAAO, oBAAO, QAAP, C; WACf, IAFE, KAeF, O; QAAK, uBIE  
7DqD, YkE6D7C, QIE7D6C, CNirD1C, QwEpnD6B, oBxEonD7B, C;; QwEnnDR, uBAAoC, cAA5B, oBAA4B, EAAV,  
CAAU, EAAP, KAAO, CAAY, QIE9DE, YkE8DK, QIE9DL, CkE8DF, EAA4C, cAAN, oBAAM, EAAY, KAAZ, EAAM  
B, SAAAnB, CAA5C, C;; MAG5D, qC; MACA, OAAO, I; K; 2CAGX, iB; MACI, qB; MACA, oBAAW, KAAX, C; MACA, q  
C; MACA, OAAW, UAAS, sBAAb, GACG, oBAAY, MADf, GAGG, oBAAY, QAAO, KAAP, EAAC, CAAd, CAAIB, CA  
AmC, CAAnC, C; K; uCAGR, mB; MAEkB, Q; MADd, qB; MACc, 2B; MAAd, mD; QACI, IAAI, 4BAAM, KAAAN, GAAg  
B, OAAhB, CAAJ, C; UACU, oBAAY, QAAO, KAAP, EAAC, CAAd, C; UACIB, qC; UACA, OAAO, I;; MAGf, OAAO, K;  
K; 8CAGX, 8B; MACI, qB; MACA, qC; MACM, oBAAY, QAAO, SAAP, EAAB, UAAU, SAAV, IAAIB, C; K; gCAGtB,  
Y; MACI, qB; MACA, uB9BhHuC, E; M8BiHvC, qC; K; wCAIJ, mB; MAA+C, OAAM, QAAN, oBAAM, EAAQ, OAAZ,  
C; K; 4CAErD, mB; MAAMd, OAAM, YAAN, oBAAM, EAAY, OAAZ, C; K; mCAEzD, Y; MAA0B, uBAAC, oBAAd, C;  
K; 0CAE1B, iB; MAGe, UAGL, MAHK, EAMO, M; MAPIB, IAAI, KAAM, OAAN, GAAa, SAAjB, C; QACI, OAAO, 2D;;  
MAGc, gBAAxB, eAAK, SAAL, IAAK, gBAAL, yB; MxEuwBL, UAAU, SAAV, EwEwBsC, KxEuwBtC, EAD+F, CA  
C/F, EADoH, CACpH, EADuI, gBACvI, C; MwErwBI, IAAI, KAAM, OAAN, GAAa, SAAjB, C; QACI, MAAM, SAAN, I  
AAc, 6E;; MAGIB, OAAO, K; K; kCAGX, Y; MACI, OAAO, EAAS, MAAM, MAAK, oBAAL, C; K; yCAIIB, Y; MACI, IA  
AI, yBAAJ, C; QAAgB, MAAM, oC; K; +CAG1B, iB; MACI, oCAAa, kCAAYB, SAAzB, C; MADoB, Y; K; wDAIrC, iB; M  
ACI, oCAAa, mCAA0B, SAA1B, C; MAD6B, Y; K;; IAIJ9C, +B; MAAA, mD; MAG8B, sB9BRa, E8BQb, C; MAH9B, Y; K  
; IAKA, kD; MAAA, mD; MAIkD, sB9BdP, E8BcO, C; MAJID, Y; K; IAMA, 2C; MAAA, mD; MAGqD, sBIENa, YkEMR,  
QIENQ, CkEMb, C; MAHrD, Y; K; ICrBJ, 0C; MACI, IAAI, 6BAAJ, C; QACU, KAAY, MAAK, UAAL, C;; QAEIB, UAAU  
, KAAV, EAAwC, CAAxC, EAAiD, cAAN, KAAM, CAAjD, EAA4D, eAAW, UAAX, CAA5D, C;; K; IAMiB, kD; MAAA  
, uB; QAAgB, OAAA, kBAAW, SAAQ, CAAR, EAAW, CAAX, C; O; K; IAFpD, 4C; MACI, IAAI, 6BAAJ, C; QACI, iBAA  
iB, gC; QACX, KAAY, MAAK, UAAL, C;; QAEIB, UAAU, KAAV, EAAwC, CAAxC, EAAiD, cAAN, KAAM, CAAjD, E  
AA4D, UAA5D, C;; K; IAIR, gE; MACI, IAAI, aAAY, UAAU, CAAV, IAAZ, CAAJ, C; QACI, UAAU, KAAV, EAAwC, S  
AAxC, EAAMd, UAAU, CAAV, IAAnD, EAAGe, UAAhE, C;; K; IAMiB, gC; MAAgB, OAAE, iBAAF, CAEE, EAAU, C  
AAV, C; K; IAF3C, 0B; MACI, IAAI, 6BAAJ, C; QACI, iBAAiB, gB; QACX, KAAY, MAAK, UAAL, C;; QAEIB, UAAU,  
KAAV, EAAwC, CAAxC, EAAiD, cAAN, KAAM, CAAjD, EAA4D, cAA5D, C;; K;; IAaa, kD; MAAoB, QAAC, IAAM, C  
AAP, KAAa, IAAM, CAAnB, K; K; IARzC, uC; MACI, sC; QAAiC, OAAjC, yB;; MACA, 4BAA4B, K; MAE5B, YAAy, E;  
MAGZ, iBAAC, CAAd, UAAsB, GAAtB, U; QAAiC, KAAY, MAAK, KAAL, C; MAC7C, iBAAiB, kC; MACX, KAAY, M  
AAK, UAAL, C; MACIB, mBAAC, CAAd, YAAsB, KAAM, OAA5B, Y; QACI, QAAQ, MAAM, UAAQ, CAAR, IAAN, C  
; QACR, QAAQ, MAAM, OAAN, C; QACR, IAAI, CAAC, IAAM, CAAP, OAAc, IAAM, CAAPB, KAA0B, KAAK, CAA  
nC, C; UAAsC, OAAO, K;; MAEjD, 4BAA4B, I; MAC5B, OAAO, I; K; IAIX, 2D; MACI, aAAa, gBAAMb, KAAM, OAAz  
B, O; MACb, aAAa, YAAU, KAAV, EAAiB, MAAjB, EAAYB, KAAzB, EAAGC, YAAhC, EAA8C, UAA9C, C; MACb, IA  
AI, WAAW, KAaf, C; QACI, aAAU, KAAV, OAAiB, YAAjB, M; UAA+B, MAAM, CAAN, IAAW, OAAO, CAAP, C;; K;  
IAIID, 4D; MAEI, IAAI, UAAS, GAAb, C; QACI, OAAO, K;; MAGX, aAAa, CAAC, QAAQ, GAAR, IAAD, IAAgB, CAA  
hB, I; MACb, WAAW, YAAU, KAAV, EAAiB, MAAjB, EAAYB, KAAzB, EAAGC, MAAhC, EAAwC, UAAxC, C; MAC  
X, YAAy, YAAU, KAAV, EAAiB, MAAjB, EAAYB, SAAS, CAAT, IAazB, EAAqC, GAARc, EAA0C, UAA1C, C; MAE  
Z, aAAiB, SAAS, MAAb, GAAqB, KAARb, GAAGC, M; MAG7C, gBAAGB, K; MACHb, iBAAiB, SAAS, CAAT, I; MACj  
B, aAAU, KAAV, OAAiB, GAAjB, M; QAEQ, iBAAa, MAAb, IAAuB, cAAc, GAARc, C; UACI, gBAAGB, KAAK, SAAL  
, C; UACHb, iBAAiB, MAAM, UAAN, C; UAEjB, IAAI, UAAW, SAAQ, SAAR, EAAMb, UAAAnB, CAAX, IAA6C, CAAj  
D, C; YACI, OAAO, CAAP, IAAY, S; YACZ, 6B;; YAEA, OAAO, CAAP, IAAY, U; YACZ, +B;; eAGR, iBAAa, MAAb, C;  
UACI, OAAO, CAAP, IAAY, KAAK, SAAL, C; UACZ, 6B;; UAGA, OAAO, CAAP, IAAY, MAAM, UAAN, C; UACZ, +  
B;; MAMZ, OAAO, M; K; ICrGX, 4C; MAMoB, UACM, M; MAHtB, IAAI, iBAAJ, C; QAAkB, OAAO, C; MACzB, aAAa,  
C; MACb, wBAAGB, SAAhB, gB; QAAgB, cAAA, SAAhB, M; QAEQ, oB; UAAmB, U;; UACnB, I1BFiC, MAAa, Y0BEn  
C, O1BFmC, C0BE9C, C; YAAwD, iCAAhC, OAAgC, C; iBAExD, uC; YAAmC, 2BAAR, OAAQ, C; eACnC, wC; YAAm  
C, 2BAAR, OAAQ, C; eACnC, sC; YAAmC, 2BAAR, OAAQ, C; eACnC, uC; YAAmC, 2BAAR, OAAQ, C;; YAEA, kBAA  
R, OAAQ, C;; QATvC, wB; QAYA, SAAS, MAAK, MAAL, QAAC, WAAAd, I;; MAEb, OAAO, M; K;; ICTP, uC; MAAA, 2  
C; K; 2DACI, 0B; MAA2D, sBAAU, MAAV, C; K; gEAE3D, iB; MAA6C, Q; MAAA, wEAAqB, C; K;; IAHTe, mD; MAAA



,kD;QAAA,iC;;MAAA,2C;K;;MC0BA,iC;MAKA,8B;MA6CA,0BAAMe,I;;IAzEnE,kC;MAAA,oB;MAA+B,8C; K;2CAE3B,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;uCAC/D,Y;MACI,WAAa,Q;K;uDAGjB,mB;MAAgE,O AAA,WAAa,uBAAc,OAAAd,C;K;0CAE7E,Y;MAAwE,OAAA,iCAAY,W;K;qDAEpF,mB;MACI,IAAI,iBAAS,OA AT,CAAJ,C;QACI,WAAa,cAAO,OAAQ,IAAf,C;QACb,OAAO,I;;MAEX,OAAO,K;K;wFAGY,Y;MAAQ,OAAA, WAAa,K;K;;8BA6ChD,Y;MACI,0BAAY,Q;K;0CAIhB,e;MAAmD,OAAA,0BAAY,gBAAS,GAAT,C;K;4CAE/D, iB;MAAmE,gBAAZ,0B;MAAY,c;;QvE+mDnD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP, e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IuE/mDmD,uBAAS,gBvE+mD9C,OuE/mDwD,MA AV,QvE+mD5D,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MuEhndgD,iB;K;kFAInD,Y;MACI,IAAI,+BA AJ,C;QACI,0BAAW,qB;;MAEf,OAAO,sC;K;uCAGf,Y;MAAgF,iC;K;kCAEHf,e;MAA+C,OAAA,0BAAY,WAAI ,GAAJ,C;K;oCAE3D,sB;MAAgD,OAAA,0BAAY,aAAI,GAAJ,EAAS,KAAT,C;K;qCAE5D,e;MAAYC,OAAA,0B AAY,cAAO,GAAP,C;K;+EAEvB,Y;MAAQ,OAAA,0BAAY,K;K;;IA5DID,0C;MAAA,iD;MAAuD,8B;MAvC3D, mB;MAwCQ,8BAAMB,W;MACnB,2BAAGB,WAA,Y,S;MAFhC,Y;K;IAKA,+B;MAAA,iD;MAGuB,aAAK,kEAA L,Q;MAHvB,Y;K;IAKA,4D;MAAA,iD;MAQ8D,qB;M7EpC9D,IAAI,E6EsCQ,mBAAMB,C7EtC3B,CAAJ,C;QA CI,c6EqCgC,+C;Q7EpChC,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,E6EuCQ,cAAc,C7EvCtB,CAAJ,C; QACI,gB6EsC2B,yC;Q7ErC3B,MAAM,gCAAYB,SAAQ,WAAjC,C;;M6E0BV,Y;K;IAcA,gD;MAAA,iD;MAA2C ,eAAK,eAAL,EAASB,GAATB,Q;MAA3C,Y;K;IAGA,yC;MAAA,iD;MAG8C,qB;MAC1C,KAAK,gBAAO,QAAP, C;MAJT,Y;K;IAqCJ,4B;MAK8E,gBAAnE,aAAmB,gEAAAnB,C;MAA2E,wB;MAAIF,O1EvCO,S;K;;M2EjEP,uB;; kCAyCA,mB;MACI,UAAU,gBAAI,aAAI,OAAJ,EAAs,IAAb,C;MACd,OAAO,W;K;8BAGX,Y;MACI,gBAAI,Q; K;uCAOR,mB;MAA6D,OAAA,gBAAI,mBAAY,OAAZ,C;K;gCAEjE,Y;MAAYC,OAAA,gBAAI,U;K;CAE7C,Y; MAAqD,OAAA,gBAAI,KAAK,W;K;qCAE9D,mB;MAAkD,OAAA,gBAAI,cAAO,OAAP,CAAJ,Q;K;+EAEpB,Y ;MAAQ,OAAA,gBAAI,K;K;;IA5D1C,6B;MAAA,iD;MAGoB,8B;MAZxB,mB;MAaQ,oBAAM,gB;MAJV,Y;K;IA OA,yC;MAAA,iD;MAG2C,8B;MAnB/C,mB;MAoBQ,oBAAM,eAAgB,QAAS,KAAzB,C;MACN,qBAAO,QAAP, C;MALJ,Y;K;IAQA,4D;MAAA,iD;MAQ2D,8B;MAhC/D,mB;MAiCQ,oBAAM,eAAgB,eAAhB,EAAiC,UAAjC, C;MATV,Y;K;IAYA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAASB,GAATB,Q;MAA3C,Y;K;IAEA,oC;MAAA,iD; MAM0C,8B;MA5C9C,mB;MA6CQ,oBAAW,G;MAPf,Y;K;IAmCJ,+B;MAKuC,gBAA5B,eAAQ,eAAR,C;MAAo C,6B;MAA3C,O3ENO,S;K;I4EzD6B,uC;MAAC,kC;MAErC,oBAAkC,kB;MACIC,sBAAyB,C;K;2EAHY,Y;MAA A,8B;K;2FAGrC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;gDAGA,sB;MACI,eAAe,aAAS,qBAA,Y,GAAZ,C;MA CxB,mBAAMB,6BAAsB,QAATB,C;MACnB,IAAI,oBAAJ,C;QAEI,kBAAW,QAAX,IAAuB,mCAAY,GAAZ,EA AiB,KAAjB,C;;QAEvB,IAAI,6BAAJ,C;UAEL,YAA+B,Y;UAC/B,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GA AIB,CAAb,C;YACI,OAAO,KAAM,gBAAS,KAAT,C;;YAEb,kBAAW,QAAX,IAAuB,CAAQ,KAAR,EAae,mCA AY,GAAZ,EAaiB,KAAjB,CAaf,C;YACvB,6B;YACA,OAAO,I;;UAIX,YAAuC,Y;UACvC,cAAkB,wBAAN,KA AM,EAaiB,GAAjB,C;UACIB,IAAI,eAAJ,C;YACI,OAAO,OAAM,gBAAS,KAAT,C;;UAEX,KAA,Y,MAAK,mC AAY,GAAZ,EAaiB,KAAjB,CAAL,C;;MAG1B,6B;MAEA,OAAO,I;K;iDAGX,e;MAEuB,Q;MADnB,eAAe,aAA S,qBAA,Y,GAAZ,C;MACL,oCAAsB,QAATB,C;MAAA,iB;QAAMC,OAAO,I;;MAA7D,mBAAMB,I;MACnB,IAA I,6BAAJ,C;QACI,YAAgC,Y;QAChC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,CAAb,C;U5BzDR,O4B 0D6B,iB5B1DvB,C4B0DmC,Q5B1DnC,C;U4B2DM,6B;UACA,OAAO,KAAM,M;;UAEb,OAAO,I;;QAGX,YAA uC,Y;QACvC,8BAAc,KAAAd,iB;UACI,cAA,Y,MAAM,KAAN,C;UACZ,IAAI,aAAS,gBAAO,GAAP,EAAY,OAA M,IAAIB,CAAb,C;YACI,IAAI,KAAM,OAAN,KAAc,CAAIB,C;cACU,KAAN,UAA2B,C;c5BtE/C,O4BwEqC,iB5 BxE/B,C4BwE2C,Q5BxE3C,C;;c4B2EoB,KAA,Y,QAAsB,QAAP,EAAC,CAAd,C;;YAEtB,6B;YAEA,OAAO,OAA M,M;;;MAIzB,OAAO,I;K;0CAGX,Y;MACI,oBAAa,kB;MACb,YAAO,C;K;mDAGX,e;MAAYC,uBAAS,GAAT, S;K;8CAEzC,e;MAA+B,Q;MAAA,+BAAS,GAAT,8B;K;+CAE/B,e;MACuB,Q;MAAA,oCAAsB,aAAS,qBAA,Y, GAAZ,CAA/B,C;MAAA,iB;QAAoD,OAAO,I;;MAA9E,mBAAMB,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;Q AChC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,CAAb,C;UACI,OAAO,K;;UAEP,OAAO,I;;QAGX,YA AuC,Y;QACvC,OAAa,wBAAN,KAAM,EAaiB,GAAjB,C;;K;uDAlrB,0B;MACI,sB;;Q7F+nCY,Q;QAAhB,iD;UA AgB,cAAhB,e;UAAsB,I6F/nCK,aAAS,gB7F+nCA,O6F/nCa,IAAb,M7F+nCd,C;YAAwB,qBAAO,O;YAAP,uB;; QAC9C,qBAAO,I;;M6FhoCH,yB;K;IAIO,8E;MAAA,wD;MACH,aAA,Y,E;MAEZ,YAA0B,MAAA,MAAK,qCAA L,C;MACvC,gBAAe,E;MAEf,oBAA4B,I;MAC5B,eAAc,K;MACd,iBAAgB,E;MAChB,iBAAqC,I;K;yEAErC,Y; MACI,IAAI,6BAAwB,YAA5B,C;QACI,gBAAqB,iBAAqD,O;QAC1E,IAAI,4DAAC,SAaIB,C;UACI,OAAO,C;;

MAGf,IAAI,yDAAa,SAAK,OAAtB,C;QACI,oBA Ae,2CAAW,UAAK,aAAL,CAAX,C;QACf,eAAU,iC;QACV,iB  
AAy,C;QACZ,OAAO,C;;QAEp,oBA Ae,I;QACf,OAAO,C;;K;mEAlf,Y;MACI,IAAI,eAAS,EAAb,C;QACI,aAAQ,  
oB;MACZ,OAAO,eAAS,C;K;gEAGpB,Y;MAEoB,Q;MADhB,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACN,  
IAAI,YAAJ,C;QACZ,yBAAqD,cAArD,C;;QAEa,OAAb,iB;;MAHJ,oB;MAKA,iBA AiB,S;MACjB,aAAQ,E;MAC  
R,OAAO,S;K;kEAGX,Y;M/E/CR,I+EgDyB,c/EhDrB,QAAJ,C;QACI,cAhByB,0B;QAIbZB,MAAM,6BAAsB,OA  
AQ,WAA9B,C;;M+E+CE,6BAAYB,cAAO,6BAAY,IAAnB,C;MACzB,iBAAY,I;MAEZ,uC;K;;6CAtdZ,Y;MAEI,  
2D;K;4DAyDJ,oB;MACI,mBAAmB,kBAAW,QAAX,C;MACnB,OAAW,iBA AiB,SAArB,GAAgC,IAAhC,GAA0  
C,Y;K;;;wCCtKrD,Y;MACI,aAAR,MAAM,OAAe,CAAP,IAAO,C;MAEb,OAAO,KAAP,IAAgB,C;M7BXpB,O6  
BYqB,M7BZf,C6BYuB,K7BZvB,C;M6BaF,OAAO,M;K;;ICNuB,qC;MAAC,kC;MAEnC,oBA AkC,kB;MACIC,sB  
AAyB,C;K;yEAHU,Y;MAAA,8B;K;yFAGnC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;iDAWA,e;MACI,IAAI,0  
BAAJ,C;QAAoB,OAAO,K;MAC3B,OAAO,kBAAW,GAAX,MAAoB,S;K;4CAG/B,e;MACI,IAAI,0BAAJ,C;QA  
AoB,OAAO,I;MAC3B,YAAy,kBAAW,GAAX,C;MACZ,OAAW,UAAU,SAArB,GAAgC,KAAhC,GAA2D,I;K;8  
CAI/D,sB;MjFVA,IAAI,EiFWQ,uBjFXR,CAAJ,C;QACI,cAda,qB;QAEb,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mi  
FUN,eAAe,kBAAW,GAAX,C;MACf,kBAAW,GAAX,IAAkB,K;MAEIB,IAAI,aAAa,SAAJB,C;QACI,6B;QAEA,  
OAAO,I;;QAGP,OAAO,Q;;K;+CAIf,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,I;MAC3B,YAAy,kBAAW,GAAX,  
C;MACZ,IAAI,UAAU,SAAd,C;Q9BnDJ,O8BoDyB,iB9BpDnB,C8BoD+B,G9BpD/B,C;Q8BqDE,6B;QAEA,OAA  
O,K;;QAGP,OAAO,I;;K;wCAkf,Y;MACI,oBA Aa,kB;MACb,YAAO,C;K;IAKA,0E;MAAA,oD;MACH,cAAkC,M  
AAa,MAAK,mCAAL,C;MAC/C,kBA A4B,qBAAL,WAAK,C;MAC5B,iBAA+B,I;K;iEAE/B,Y;MAAkC,OAAA,e  
AAS,U;K;8DAE3C,Y;MAIuB,gB;MAHnB,UAAU,eAAS,O;MACnB,iBAAU,G;MAES,+E;MAAnB,OAAO,iD;K;  
gEAGX,Y;MAEkC,UAA9B,M;MAAA,oC;MAA8B,YAAa,c;MjFchD,uB;MAeP,IAfoB,KAehB,QAAJ,C;QACI,cA  
hByB,0B;QAIbZB,MAAM,6BAAsB,OAAQ,WAA9B,C;;QAEAn,sBAnBgB,K;;MiFde,oBAAO,sFAAP,C;K;;2CAjB  
nC,Y;MACI,yD;K;IAqBkd,0F;MAAA,8B;MAAA,oD;K;kHAC9B,Y;MAAQ,uB;K;oHACN,Y;MAAQ,6CAAuB,g  
BAAvB,C;K;2EAE9B,oB;MAAwC,OAAA,2BA AuB,aAAI,gBA AJ,EAAS,QAAT,C;K;qEAE/D,Y;MAA+B,OAA  
A,mCAAY,uBAAc,IAAd,C;K;qEAC3C,Y;MAAkC,OAAA,mCAAY,uBAAc,IAAd,C;K;mEAC9C,iB;MAA4C,OA  
AA,mCAAY,qBAAY,IAAZ,EA AkB,KAAIB,C;K;;gDAR5D,e;MAAsD,iE;K;;;MCItD,sBAOsC,I;MA6CtC,yB;MA  
OA,4BA AkC,K;;IArIE,sD;MAZpC,oB;MAYyD,0CAAqC,GAArC,EA A0C,KAA1C,C;MACrD,oBA AuC,I;MACv  
C,oBA AuC,I;K;wDAEvC,oB;MACI,WAAmB,iB;MACnB,OAAa,mEAAS,QAAT,C;K;;IAIrB,wC;MAAA,oB;MA  
A+B,8C;K;IAE3B,sD;MAAA,oB;MACI,cACsC,I;MAEtC,cACsC,I;MAGIC,cAAO,iC;K;6DAIX,Y;MACI,OAAO,  
gBAAS,I;K;0DAGpB,Y;MAEI,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MAEtB,cAAc,0B;MACd,cAAO,O;MA  
Ca,gBAAb,OAAQ,a;;MAAf,c/E0DS,S+E1DoB,KAAO,iC/E0DzC,GAAqB,SAArB,GAA+B,I;M+EzD1B,OAAO,O  
;K;4DAGX,Y;MIFwBR,IAAI,EkFvBc,eAAQ,IIFuBtB,CAAJ,C;QACI,cAdW,e;QAEX,MAAM,6BAAsB,OAAQ,W  
AA9B,C;;MkFxBE,WAAc,iB;MAGP,oCAAP,0BAAO,C;MACP,gCAAI,cAAO,0BAAO,IAAd,C;MAEJ,cAAO,I;K  
;;iDAIf,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;6CAC/D,Y;MACI,WAAmB,Q;K;6DAGvB,mB;MAAgE,OA  
AA,WAAmB,uBAAc,OAAd,C;K;gDAEnF,Y;MAAwE,qD;K;2DAExE,mB;MACI,qB;MACA,IAAI,iBAAS,OAAT  
,CAAJ,C;QACI,WAAmB,cAAO,OAAQ,IAAf,C;QACnB,OAAO,I;;MAEX,OAAO,K;K;8FAGY,Y;MAAQ,OAAA  
,WAAmB,K;K;sDAEID,Y;MAAsC,WAAmB,iB;K;;iDAa7D,qB;MIFrBA,IAAI,EkF0BM,0BAAQ,IAAR,IAAgB,0  
BAAQ,IIF1B9B,CAAJ,C;QACI,cAdW,e;QAEX,MAAM,6BAAsB,OAAQ,WAA9B,C;;MkF0BN,YAAy,mB;MAC  
Z,IAAI,SAAS,IAAb,C;QACI,sBAAO,S;QACP,yBAAO,S;QACP,yBAAO,S;;QAGK,YAAa,KAAM,a;QIFIBhC,uB  
;QAEp,IAfoB,KAehB,QAAJ,C;UACI,gBAhByB,0B;UAIbZB,MAAM,6BAAsB,SAAQ,WAA9B,C;;UAEN,sBAnB  
gB,K;;QkFkBZ,+B;QAEA,yBAAO,K;QACP,yBAAO,K;QAEp,qBAAa,S;QACb,qBAAa,S;;K;+CAIrB,qB;MAIL,I  
AAI,SAAK,aAAL,KAAc,SAAlB,C;QAEI,sBAAO,I;;QAEp,IAAI,wBAAS,SAAb,C;UAEI,sBAAO,sB;;QAEX,qD  
AAc,sB;QACd,qDAAc,sB;;MAEIB,yBAAO,I;MACP,yBAAO,I;K;oCA8CX,Y;MAEI,qB;MACA,4BA Aa,I;MACb,  
OAAO,I;K;oCAGX,Y;MACI,qB;MACA,kBA AI,Q;MACJ,sBAAO,I;K;gDASX,e;MAAmD,OAAA,kBA AI,mBAA  
Y,GAAZ,C;K;kDAEvD,iB;MACiC,Q;MAAA,0B;MAAA,iB;QAAQ,OAAO,K;;MAA5C,WAA6B,I;;QAEzB,IAAI,  
OAAA,IAAK,MAAL,EA Ac,KAAAd,CAAJ,C;UACI,OAAO,I;;QAEX,OAAO,cAAA,IAAK,aAAL,C;;MACF,iBAA  
S,mBAAT,C;MACT,OAAO,K;K;6CAIX,Y;MAAoF,uC;K;wCAEpF,e;MAAmD,Q;MAAJ,QAAI,OAAJ,kBA AI,W  
AAI,GAAJ,CAAJ,6B;K;0CAE/C,sB;MACI,qB;MAEA,UAAU,kBA AI,WAAI,GAAJ,C;MACd,IAAI,OAAO,IAAX  
,C;QACI,eAAe,mCAAW,GAAX,EA AgB,KAAhB,C;QACf,kBA AI,aAAI,GAAJ,EAAS,QAAT,C;QACK,wBAAT,

QAAS,C;QACT,OAAO,I;QAEP,OAAO,GAAL,gBAAS,KAAT,C;K;2CAInB,e;MACI,qB;MAEA,YAA Y,kBAAL,cAAO,GAAP,C;MACHb,IAAI,SAAS,IAAb,C;QACU,sBAAN,KAAM,C;QACN,OAAO,KAAM,M;MAEjB,OAA O,I;K;qFAGmB,Y;MAAQ,OAAA,kBAAL,K;K;6CAE1C,Y;MACI,IAAI,yBAAJ,C;QAAgB,MAAM,oC;K;;IANg1 B,mC;MAAA,uD;MAGuB,qB;MA9J3B,yB;MA+JQ,sBAAM,gB;MAJV,Y;K;IAOA,iD;MAAA,uD;MAAoD,qB;M AlKxD,yB;MAoKc,Q;MAAN,sBAAM,+D;MAFV,Y;K;IAKA,kE;MAAA,uD;MAQ8D,eAAM,eAAN,EAAuB,UA AvB,Q;MA/KIE,yB;MAGLQ,sBAAM,gB;MATV,Y;K;IAYA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAA sB,GA AtB,Q;MAA3C,Y;K;IAEA,+C;MAAA,uD;MAG2C,qB;MAxL/C,yB;MAyLQ,sBAAM,gB;MACN,KA AK,gBAAO ,QAAP,C;MALT,Y;K;IA6EJ,kC;MAKwD,gBAA7C,qBAAYB,eAAzB,C;MAAqD,wB;MAA5D,O/EjMO,S;K;;;oC gFvCP,Y;MAEK,Q;MAA8B,CAA9B,2EAA8B,S;MAC/B,OAAO,I;K;6CAGX,Y;MAA+C,gBAAL,iB;K;;IAhCnD, wC;MAAA,uD;MAAmD,eAAM,GAAN,Q;MAPvD,yB;MAOI,Y;K;IAEA,qC;MAAA,uD;MAGuB,eAAM,oBAAN ,Q;MAZ3B,yB;MASI,Y;K;IAKA,+C;MAAA,uD;MAG8C,eAAM,oBAAN,Q;MAjBID,yB;MAkBQ,qBAAO,QAA P,C;MAJJ,Y;K;IAOA,kE;MAAA,uD;MAQ8D,eAAM,qBAAsB,eAAtB,EAAuC,UAAvC,CAAN,Q;MA7BIE,yB;M AqBI,Y;K;IAUA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAA sB,GAAtB,Q;MAA3C,Y;K;IAgBJ,qC;MAK mD,gB AAxC,mBAAc,qBAAd,C;MAAgD,6B;MAAvD,OhFoBO,S;K;;;kFiFzEX,uB;MAQI,OAAO,O;K;ICXX,sB;K;mC ACI,Y;MACI,mBAAM,IAAN,C;K;2CAGJ,mB;MACI,mBAAM,OAAN,C;MACA,c;K;iCAKJ,Y;K;;IAKuB,oC;M AA8B,qB;MAA7B,gC;K;2CACxB,mB;MAEI,oBA+DyC,OA/Dd,OA+Dc,C;MA9DzC,iBAaA,OAAM,aAAN,C;K ;IAIrB,8B;MAEoC,qB;K;iDACHC,mB;MACI,OAAQ,KAAI,OAAJ,C;K;mDAGZ,mB;MACI,OAAQ,KAAI,OAAJ, C;K;2CAGZ,Y;MACI,OAAQ,KAAI,EAAJ,C;K;;IAIhB,0B;MAEqC,qB;MACjC,cAAa,E;K;6CAEb,mB;MACI,eA oCyC,OApCxB,OAoCwB,C;K;qCAjC7C,Y;MACI,cAAS,E;K;;IAIjB,sC;MAE4C,yB;K;yDACxC,mB;MACI,QA w ByC,OAxB1B,OAwB0B,C;MAvBzC,QAAQ,CxEqJoF,awErJhE,IxEqJgE,EwErJ1D,CxEqJ0D,C;MwEpJ5F,IAAI, KAAK,CAAT,C;QACI,4BAAU,CxE+J0E,WwE/J9D,CxE+J8D,EwE/J3D,CxE+J2D,C;QwE9JpF,Y;QACA,IAAI,C xE0JiE,WwE1JrD,IAAI,CAAJ,IxE0JqD,C;;MwExJzE,4BAAU,C;K;iDAGd,Y;MACI,OAAQ,KAAI,WAAJ,C;MA CR,cAAS,E;K;;IAWjB,yB;MACiD,cAAa,KAAb,C;K;IAEjD,mB;MAEL,MAAO,U;K;IAGX,4B;MAEI,MAAO,iB AAQ,OAAR,C;K;IAGX,wB;MAEI,MAAO,eAAM,OAAN,C;K;IAGX,kB;MACqC,MAAM,qCAA8B,sCAA9B,C; K;IAE3C,wB;MAC4C,MAAM,qCAA8B,4CAA9B,C;K;ICIGID,mD;MACI,0B;MASA,gBAA2B,a;K;2FAFvB,Y;M AAQ,OAAA,eAAS,Q;K;oDAIrB,kB;MACI,UAAU,IAAK,S;MAEX,YAAQ,2CAAR,C;QACI,gBAAc,MAAO,M; WAEzB,YAAQ,yBAAR,C;QACI,gBAAc,yC;QACd,eAAS,oBAAW,MAAX,C;;QAEL,MAAM,6BAAsB,iBAAtB, C;K;4CAItB,Y;MAOW,Q;MALP,IAAI,kBAAW,2CAAf,C;QACI,gBAAS,yB;QACT,OAAO,yB;;MAEX,aAAa,IA AK,S;MAEd,eAAW,yCAAX,C;QAAsB,gC;WACtB,0C;QAA4B,MAAM,MAAO,U;;QACjC,a;MAHZ,W;K;;IA7B J,gD;MAAA,0D;MACyD,6BAAK,QAAL,EAAe,2CAAf,C;MADzD,Y;K;;;ICRA,2C;MAAA,+D;MAAuB,iC;MA F3B,iC;MAEI,Y;K;IACA,sD;MAAA,+D;MAAuC,6BAAM,OAAN,Q;MAH3C,iC;MAGI,Y;K;IACA,6D;MAAA,+ D;MAAmD,kCAAM,OAAN,EAAe,KAAf,C;MAJvD,iC;MAIL,Y;K;IACA,oD;MAAA,+D;MAAiC,6BAAM,KAA N,Q;MALrC,iC;MAKI,Y;K;IxC4CJ,yE;MASI,sC;MAAA,4C;K;IATJ,iGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,0D AaQ,kB;MACI,wBAAW,MAAX,C;K;IAdZ,sF;IyC5C2E,0C;M1CkKhE,Q;MADP,e0ChKA,M1CgKA,C;MACO,Q 0CjKP,M1CiKO,+D;M0ChKX,W;K;;+FCuHA,gB;MACI,aAAa,IAAb,MAAa,E;MACb,KAAK,MAAL,C;MACA, OAAO,M;K;wFC3HX,yB;MAAA,uD;MAAA,wC;QAWoG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB, C;O;KAX1G,C;wFAaA,yB;MAAA,uD;MAAA,wC;QAWoG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB, C;O;KAXzG,C;8ECbA,yB;MAAA,6C;MAAA,sC;QAOyD,OAAK,SAAL,SAAK,EAA Y,QA AZ,C;O;KAP9D,C;8E ASA,yB;MAAA,6C;MAAA,wC;QAWkE,OAAK,SAAL,SAAK,EAAa,UAAb,S;O;KAXvE,C;oFAaA,yB;MAAA, mD;MAAA,wC;QAWqE,OAAK,YAAL,SAAK,EAAgB,UAAhB,S;O;KAX1E,C;kFCZI,yB;MAAA,iD;MAAA,4B; QA Ae,OAAK,WAAL,SAAK,C;O;KAApB,C;wFAYA,yB;MAAA,uD;MAAA,4B;QA Ae,OAAK,cAAL,SAAK,C;O ;KAApB,C;IC5BJ,gC;MAAoE,gCAAqB,OAAR,B,C;K;IAEIC,uC;MAAC,wB;K;iDAC/B,iB;MACI,eAAQ,KAAR,C ;K;8CAGJ,Y;MAAyC,iCAAuB,cAAvB,M;K;;ICCO,6C;MAAA,8B;MAAS,uB;K;8FACIC,Y;MAAQ,OAAA,gBA AY,O;K;mDAE3C,iB;MACI,IADoC,KACpC,IAAG,CAAH,IADoC,KACpC,IAAM,sBAAN,C;QAD8B,OACX,gB AAY,MAAK,KAAL,C;;QACvB,MAAM,8BAA0B,WAAQ,KAAR,6BAAmC,sBAAnC,MAA1B,C;K;;IARtB,8B; MAGoD,4C;K;wECFpD,yB;MAAA,uC;MAAA,4B;QAOsC,MAAL,SAAK,C;O;KAPtC,C;kFASA,yB;MAAA,iD; MAAA,kC;QAWuD,OAAK,WAAL,SAAK,EAAc,IAAd,C;O;KAX5D,C;+ECfA,qB;MAI8C,gB;K;iFAE9C,qB;M AIsE,OAAK,S;K;kFAE3E,qB;MAMyE,gB;K;IAEzE,6B;MAiBa,UAPF,M;MAFP,QAAc,S;MAGV,cAAK,UAAAL,

U;QACI,mBAAK,UAAAL,G;;QACJ,I/CzBqC,MAAa,Y+CyBvC,C/CzBuC,C+CyBID,C;UAC6B,8BAAzB,CAAyB,C;;UAGN,UAAIB,uDAakB,Y;;MAP3B,a;K;IC9BJ,2B;MAEI,MAAM,yBAAqB,OAARb,C;K;IAGV,sB;MAEI,MAAM,uBAAMb,cAAAnB,C;K;IAGV,2B;MAEI,MAAM,6BAAsB,OAAtB,C;K;IAGV,iC;MAEI,MAAM,4CAAqC,uBAAqB,YAArB,8BAArC,C;K;ICIBV,8B;MC8CW,kB1GqBiD,oB;M0GM9C,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAakB,sBAAY,GAAZ,C;QAKFiD,U;QAJFnE,W1GuKJ,a0GvKgB,G1GuKhB,EyG10oB,CCmEkC,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAiFhD,GDpJrC,CCoJqC,GAA6B,UAjFjC,WaFiC,6DDpJnD,IAAM,CAAN,IzG0OpB,C;;MyG10A,OCqEO,W;K;IC3EqC,gD;MAAC,oC;K;;;IC0CjD,qB;MAK0B,Q;MADtB,UAAmB,E;MACnB,wBAAsB,KAAtB,gB;QAAsB,aAAA,KAAtB,M;QAAK,IAAC,0BAAD,EAAO,2B;QACR,IAAI,IAAJ,IAAY,K;;MAEhB,OAAO,G;K;IAGX,+B;MAMgB,Q;MADZ,WAA0B,MAAa,MAAK,KAAL,C;MACvC,wBAAY,IAAZ,gB;QAAY,UAAA,IAAZ,M;QACI,IAAU,KAAy,gBAAe,GAAf,CAAtB,C;UACI,UAAK,GAAL,IAAY,MAAM,GAAN,C;;;MAGpB,OAAO,S;K;qEC5DX,yB;MAAA,iB;MAAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;uEASA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;yEAWA,yB;MAAA,iB;MAAA,uB;QAKb+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAIB1E,C;uEAoBA,yB;MAAA,iB;MAAA,oB;QAUMD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;uEAYA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAUMD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAYoD,OAAA,MAAW,OAAM,CAAN,C;O;KAZ/D,C;yEAca,yB;MAAA,iB;MAAA,oB;QAYoD,OAAA,MAAW,OAAM,CAAN,C;O;KAZ/D,C;yEAca,yB;MAAA,iB;MAAA,oB;QAaoD,OAAA,MAAW,OAAM,CAAN,C;O;KAb/D,C;yEAea,yB;MAAA,iB;MAAA,uB;QAS+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAT1E,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAQmD,OAAA,MAAW,MAAK,CAAL,C;O;KAR9D,C;qEAUA,yB;MAAA,iB;MAAA,oB;QAUKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAV7D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAcoD,OAAA,MAAW,OAAM,CAAN,C;O;KAd/D,C;IAGbA,sB;MAcI,IAAI,QAAQ,GAAR,IAAe,SAAQ,GAA3B,C;QAAgC,OAAO,wCAA0,I;MAC9C,OAAO,IAAW,KAAL,CAAJ,CAAX,GAAoB,IAAW,KAAL,IAAJ,C;K;mEAG1C,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAAW,KAAL,CAAJ,C;O;KAX5D,C;yEAaA,yB;MAAA,iB;MAAA,oB;QAOoD,OAAA,MAAW,OAAM,CAAN,C;O;KAP/D,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOMD,OAAA,MAAW,MAAK,CAAL,C;O;KAP9D,C;uEASA,yB;MAAA,iB;MAAA,oB;QAgbmD,OAAA,MAAW,OAAM,CAAN,C;O;KAhB9D,C;uEakBA,yB;MAAA,iB;MAAA,oB;QAUMD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAUoD,OAAA,MAAW,OAAM,CAAN,C;O;KAV/D,C;+EAYA,yB;MAAA,iB;MAAA,oB;QAUuD,OAAA,MAAW,OAAM,CAAN,C;O;KAVIE,C;IAYA,kB;MAQI,IAAI,IAAI,GAJ,KAaw,GAAf,C;QACI,OAAO,IAAW,OAAM,CAAN,C;;MAEtB,YAzBgD,MAAW,OAYBzC,CAZByC,C;MA0B3D,OAAW,QAAQ,CAAR,KAAa,GAAxB,GAA6B,KAA7B,GAtC+C,MAAW,MAcB,CAtCa,C;K;qEAyC9D,yB;MAAA,iB;MAAA,oB;QAUKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAV7D,C;uEAYA,yB;MAAA,iB;MAAA,oB;QAWmD,OAAA,MAAW,MAAK,CAAL,C;O;KAX9D,C;wEAca,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAAL,CAAJ,EAAO,CAAP,C;O;KAPxE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAAL,CAAJ,EAAO,CAAP,C;O;KAPxE,C;qEAWA,yB;MAAA,iB;MAAA,+B;QAAYD,OAAA,MAAW,KAAL,SAAJ,EAAU,CAAV,C;O;KAbpE,C;uEAea,yB;MAAA,iB;MAAA,+B;QAOSD,OAAA,MAAW,KAAL,SAAJ,EAAy,CAAZ,C;O;KAPjE,C;iGAmBsD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,KAAL,SAAJ,C;O;KAAAnB,C;+EAaT,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,MAAK,SAAL,C;O;KAAAnB,C;iFAE7C,yB;MAAA,6C;MAAA,kC;QAK8D,OAAK,SAAL,SAAK,EAAC,IAAD,C;O;KALnE,C;IAKbqC,4B;MACjC,gBAAO,CAAP,C;QADyC,OACrB,QAAP,CAAC,SAAM,C;WACpB,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QAFyC,OAEW,S;WACpD,kBAAQ,wCAA0,UAAf,C;QAHyC,OAGb,YAAY,SAAL,SAAK,C;;QAHc,OAI5B,OAAL,SAAK,CAAL,GAAGB,S;K;IAG5B,2B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QADwC,OACY,S;WACpD,kBAAQ,GAAR,C;QAFwC,OAEzB,wCAA0,U;;QACP,WAAc,UAAAL,SAAK,CAAL,yBAAuB,YAAO,CAAX,GAAc,CAAd,GAAqB,EAAxC,E;QAHgB,OjDhb6B,MAAa,gBAAe,IAAf,C;;K;liDsbtF,6B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QAD0C,OACU,S;WACpD,kBAAQ,GAAR,C;

QAF0C,OAE3B,CAAC,wCAAO,U;;QACR,WAAc,UAAL,SAAK,CAAL,yBAAuB,YAAO,CAAX,GAAC,EAAd,GAAsB,CAAzC,E;QAHkB,OjD1b2B,MAAA,gBA Ae,IAAf,C;;K;liDictF,oC;MAUI,IAAK,QAAL,SAAK,CAAL,IAAmB,QA AH,EAAG,CAAnB,C;QADuD,OACzB,wCAAO,I;WACrC,WAAM,SAAN,C;QAFuD,OAEzC,E;WACd,SAAK,SAAL,C;QAHuD,OAGrC,OAAL,SAAK,C;;QAHqC,OAI1B,SAAL,SAAK,C;K;IAIjC,+B;MAYI,uB;QAAW,MAAM,gCAAyB,yBAAzB,C;WACjB,gBAAO,UAAP,C;QAFyC,OAEjB,U;WACxB,gBAAO,WAAP,C;QAHyC,OAGjB,W;;QAHiB,OAIv,YAAvB,IAAW,OAAM,SAAN,CAAY,C;K;IAGnC,gC;MAYI,uB;QAAW,MAAM,gCAAyB,yBAAzB,C;WACjB,oD;QAF2C,+B;WAG3C,oD;QAH2C,+B;;QAAA,OAIz,uBAAvB,IAAW,OAAM,SAAN,CAAY,C;K;uEASnC,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;yEASA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;2EAWA,yB;MAAA,iB;MAAA,uB;QAKb4D,OAAA,MAA6C,OAA1B,CAA0B,EA AZ,CAAY,C;O;KAIbZG,C;yEAoBA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;yEAYA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAYkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAZjF,C;2EAcA,yB;MAAA,iB;MAAA,oB;QAYkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAZjF,C;2EAcA,yB;MAAA,iB;MAAA,oB;QAakD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAbjF,C;2EAeA,yB;MAAA,iB;MAAA,uB;QAS4D,OAAA,MAA6C,OAA1B,CAA0B,EA AZ,CAAY,C;O;KATzG,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAQiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAR/E,C;uEAUA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAcKd,OAAA,MAA+B,OAAZ,CAAY,C;O;KAdjF,C;uEAgBA,yB;MAAA,mC;MAAA,OB;QAc6D,OAAmC,IAA7B,CAA6B,EA AZ,IAAY,C;O;KAdhG,C;qEAgBA,yB;MAAA,iB;MAAA,oB;QAW+C,OAAA,MAA6B,KAAZ,CAAY,C;O;KAX5E,C;2EAaA,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAPjF,C;yEASA,yB;MAAA,iB;MAAA,oB;QAOiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAP/E,C;yEASA,yB;MAAA,iB;MAAA,oB;QAgBiD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAhBhF,C;yEAKBA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAVjF,C;iFAYA,yB;MA3gBA,iB;MA2gBA,oB;QAUqD,OA3gBE,MAAW,OA2gBF,CA3gBE,C;O;KAigBIE,C;2EAYA,yB;MAAA,uC;MAAA,oB;QAQkD,OA AoB,MAAZ,CAAY,C;O;KARtE,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAX/E,C;wEAeA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;sEAYA,yB;MAAA,iB;MAAA,+B;QAAsD,OAAA,MAA8C,KAA1B,SAA0B,EA AZ,CAAY,C;O;KAbpG,C;uEAeA,yB;MAAA,iB;MAAA,+B;QAO oD,OAAA,MAA8C,KAA1B,SAA0B,EA AZ,CAAY,C;O;KAPIG,C;kGAmBoD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAgC,KAAZ,SAAY,C;O;KAAxC,C;gFAaT,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAiC,MAAZ,SAAY,C;O;KAAzC,C;gFAE3C,yB;MAAA,6C;MAAA,kC;QAO8D,OAA0C,SAArC,SAAqC,EA AZ,IAAY,C;O;KAPxG,C;iFASA,yB;MAAA,6C;MAAA,kC;QAK4D,OAA0C,SAArC,SAAqC,EA AZ,IAAY,C;O;KALtG,C;oFAQA,yB;MAAA,iD;MAAA,4B;QAYmD,OA AW,WAAX,SAAW,C;O;KAZ9D,C;sFAcA,yB;MAAA,mD;MAAA,4B;QAYqD,OA AW,YAAX,SAAW,C;O;KAZhE,C;IAoBa,kB;MAUqC,OAAI,IAAI,CAAR,GAAY,CAAC,CAAD,OAA M,CAAIB,GAA0B,C;K;wEAE/D,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;wEAOA,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;mGAIbGd,yB;MAAA,mC;MAAA,4B;QAAQ,WAAL,SAAJ,C;O;KAAR,C;IAShB,+B;MAC5B,gBAAO,CAAP,C;QADoC,OACxB,E;WACZ,gBAAO,CAAP,C;QAFoC,OAExB,C;;QAFwB,OAG5B,C;K;IAKZ,kB;MASuC,OAAI,eAAI,CAAR,GAAY,CAAD,aAAX,GAAMB,C;K;wEAE1D,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;wEAE1E,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;mGAYxB,yB;MAAA,mC;MAAA,4B;QAAQ,WAAL,SAAJ,C;O;KAAR,C;IASjB,+B;MAC7B,2BAAO,CAAP,C;QADqC,OACzB,E;WACZ,2BAAO,CAAP,C;QAFqC,OAEzB,C;;QAFyB,OAG7B,C;K;IC1kCZ,4B;MAI4C,qBAAQ,S;K;IAEpD,4B;MAI2C,qBAAQ,S;K;IAEnD,+B;MAGiD,qBAAQ,wCAAO,kBAAf,IAAoC,cAAQ,wCAAO,kB;K;IAEpG,iC;



K;kEACA,Y;MAAA,+B;K;yEACA,Y;MAAA,sC;K;iCAEA,iB;MACI,0CACQ,wBAAc,KAAM,WAApB,CADR,I  
AC0C,uBAAa,KAAM,UAAAnB,CAD1C,IAC0E,0BAAoB,KAAM,iB;K;mCAExG,Y;MACI,SAAC,CAAW,SAAX,  
eAAW,CAAX,GAAwB,EAAxB,QAAuC,SAAV,cAAU,CAAvC,IAAD,IAAsD,EAAtd,QAA4E,SAAjB,qBAAiB,C  
AA5E,I;K;mCAEJ,Y;MACKB,UACO,M;MADrB,aAAc,2D;MAEV,cAAU,IAAV,C;QAA6B,SAAX,eAAW,W;WA  
C7B,IAAA,MAAO,WAAP,S;QAAoC,SAAP,MAAO,W;;QAC5B,+B;MAHZ,2B;MAMA,WACQ,cAAU,UAAAd,G  
AAyB,EAAzB,GACe,eAAV,cAAU,EAAa,IAAb,EAAmB,GAAAnB,EAAbW,GAAxB,C;MACnB,eAAmB,qBAAJ,  
GAA5B,GAAiB,GAA+B,E;MAE9C,OAAO,iBAAiB,IAAjB,GAAwB,Q;K;;IAIvC,wB;MAAA,4B;MACI,4BAAw  
C,I;MACxC,2BAAgD,W;MACHd,kCAAyC,K;K;0FAFzC,Y;MAAA,gC;K;yFACA,Y;MAAA,+B;K;gGACA,Y;M  
AAA,sC;K;sCACA,Y;MAAkC,gB;K;;IAJtC,oC;MAAA,mC;QAAA,kB;;MAAA,4B;K;IC7BsC,oE;MACIC,0B;M  
ACA,wC;MACA,kC;MACA,oC;K;sEHA,Y;MAAA,0B;K;6EACA,Y;MAAA,iC;K;0EACA,Y;MAAA,8B;K;2EA  
CA,Y;MAAA,+B;K;4CAEA,Y;MAAkC,gB;K;;8CANtC,Y;MACI,gB;K;8CADJ,Y;MAEI,uB;K;8CAFJ,Y;MAGI,o  
B;K;8CAHJ,Y;MAII,qB;K;gDAJJ,kD;MAAA,8BACI,kCADJ,EAEL,uDAFJ,EAGI,8CAHJ,EAII,iDAJJ,C;K;4CAA  
A,Y;MAAA,c;MACI,qD;MACA,4D;MACA,yD;MACA,0D;MAJJ,a;K;0CAAA,iB;MAAA,4IACI,oCADJ,IAEL,kD  
AFJ,IAGI,4CAHJ,IAII,8CAJJ,I;K;ICAA,4B;MAAA,gC;MAEI,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAo  
E,gCAApE,C;MAEf,mBACKB,wBAAoB,MAApB,EAAGe,QAaHE,EAA0E,mCAA1E,C;MAEIB,oBACmB,+B;M  
AEnB,oBACmB,wBAAoB,OAApB,EAaKE,SAaIE,EAA6E,oCAA7E,C;MAEnB,iBACgB,wBAAoB,MAApB,EA  
A8D,MAA9D,EAA5E,iCAAtE,C;MAEhB,kBACiB,wBAAoB,MAApB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MA  
EjB,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCAApE,C;MAEf,kBACiB,wBAAoB,MAApB,EAA+D,  
OAA/D,EAAwE,kCAAxE,C;MAEjB,mBACKB,wBAAoB,MAApB,EAAGe,QAaHE,EAA0E,mCAA1E,C;MAEIB,  
kBACiB,wBAAoB,KAAPB,EAaIE,OAAjE,EAA0E,kCAA1E,C;MAEjB,mBACKB,wBAAoB,MAApB,EAAGe,Q  
AAhE,EAA0E,mCAA1E,C;MAEIB,sBACqB,wBAAoB,KAAPB,EAaKE,WAAIE,EAA+E,sCAA/E,C;MAErB,yB  
ACwB,wBAAoB,KAAPB,EAaQE,cAArE,EAaQF,yCAArF,C;MAExB,sBACqB,wBAAoB,WAApB,EAAwE,WA  
AxE,EAaQF,sCAArF,C;MAErB,sBACqB,wBAAoB,SAAPB,EAAsE,WAAIE,EAAmF,sCAAnF,C;MAErB,uBACs  
B,wBAAoB,UAApB,EAAwE,YAAxE,EAAsF,uCAAtF,C;MAEtB,qBACoB,wBAAoB,UAApB,EAAsE,UAAIE,E  
AAkF,qCAaIF,C;MAEpB,sBACqB,wBAAoB,KAAPB,EAaKE,WAAIE,EAA+E,sCAA/E,C;MAErB,uBACsB,wB  
AAoB,YAApB,EAA0E,YAA1E,EAAwF,uCAAXF,C;MAEtB,wBACuB,wBAAoB,YAApB,EAA2E,aAA3E,EAA0  
F,wCAA1F,C;K;IAMkB,qE;MAAA,qB;QAAE,OvE/DD,OuE+DU,EAAT,KAAiB,UAAjB,IAAkC,EAAY,OAAf,K  
AA0B,a;O;K;+CAJpG,iB;MAE2B,Q;MAAhB,U;MAAA,KAAGB,OAAhB,eAAGB,CAAI,KAAJ,CAAhB,U;QAAA  
,a;;QACH,aAAa,wBAAoB,QAApB,EAA+D,kBAA/D,EACoB,mDADpB,C;QAEg,eAAhB,UAAqC,M;QAHIC,SA  
IH,M;;MAJJ,a;K;IA7D+E,8C;MAAE,6B;K;IAGO,iD;MAAE,0B;K;IAME,kD;MAAE,8B;K;IAGZ,+C;MAAE,6B;  
K;IAGC,gD;MAAE,6B;K;IAGR,8C;MAAE,6B;K;IAGI,gD;MAAE,6B;K;IAGC,iD;MAAE,6B;K;IAGH,gD;MAA  
E,yB;K;IAGD,iD;MAAE,6B;K;IAGM,oD;MAAE,mC;K;IAGO,uD;MAAE,gC;K;IAGL,oD;MAAE,6B;K;IAGJ,oD  
;MAAE,6B;K;IAGE,qD;MAAE,8B;K;IAGR,mD;MAAE,4B;K;IAGJ,oD;MAAE,6B;K;IAGQ,qD;MAAE,8B;K;IA  
GC,sD;MAAE,+B;K;;IA5DvH,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;;ICCA,2B;MAEW,Q;MAAA,IAAI,KAA  
Y,SAAQ,MAAR,CAAhB,C;QACH,kBAAW,MAAX,C;;QAEA,kBAAW,MAAX,C;;MAHJ,W;K;IAOJ,8B;MAC4  
E,QAAM,QAAS,OAAf,C;aACxE,C;UADwE,OACnE,WAAW,SAAS,CAAT,CAAX,C;aACL,C;UAFwE,OAEnE,+  
B;;UAFmE,OAGhE,iB;;K;IAGZ,oC;MAEU,IAAN,I;MAAA,QxEhB0C,OwEgB3B,CAAf,C;aACI,Q;UAA6B,OAA  
jB,8BAAiB,Y;UAA7B,K;aACA,Q;UAAy,OAAI,CAAY,C/DbhC,G+DamC,CAAf,MAAkC,CAAtC,GAAyC,8BA  
AiB,SAAI,D,GAAwE,8BAAiB,Y;UAArG,K;aACA,S;UAA8B,OAAjB,8BAAiB,a;UAA9B,K;aACA,U;UAA+B,O  
AAjB,8BAAiB,eAAGB,CAAY,OAA5B,C;UAA/B,K;;UAGQ,6B;YAA5C,OAAjB,8BAAiB,kB;eACtC,0B;YAAmC  
,OAAjB,8BAAiB,e;eACnC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,yB;  
YAAkC,OAAjB,8BAAiB,c;eACIC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eAC  
pC,4B;YAAqC,OAAjB,8BAAiB,iB;eACrC,6B;;eACA,sB;YAAkC,OAAjB,8BAAiB,W;;YAE9B,kBAaKB,MAAA,  
gBAaE,CAAf,CAAkB,Y;YAE7C,oBAAGB,MAAhB,C;CAAI,D,OAAjB,8BAAiB,S;iBACjD,oBAAGB,KAAhB,C;c  
AAgD,OAAjB,8BAAiB,e;;cAE5C,cAA0B,W;cAC1B,kBAAW,OAAX,C;;;UAxBxB,K;;MAAA,W;K;IAGCJ,4B;  
MAMW,Q;MAJP,IAAI,WAAW,MAAf,C;QAA6B,OAAO,8BAAiB,Y;;MAErD,eAAsB,MAAY,W;MAE3B,IAAI,  
gBAAJ,C;QACH,IAAI,QAAS,SAAT,QA AJ,C;UACI,aAAa,qBAAiB,MAAJB,C;UACb,oBAAsB,M;UACtB,a;;UA  
ES,OAAT,QAAS,S;;QAGb,4BAAiB,MAAJB,C;;MATJ,W;K;ICrCJ,0B;MAII,sBAAY,C;K;qEAChB,4B;MAIkE,iB

AAy,KAAZ,C;K;2EAEIE,qB;MAI8D,gB;K;ICIDb,2C;MAC7C,qBAAwC,Q;K;iDAExC,Y;MACmB,Q;MAAA,yB ;MAAA,iB;QAAe,MAAM,6BAAsB,0CAAtB,C;;MAApC,eAAe,I;MACf,qBAAc,I;MACd,OAAO,QAAS,W;K;;;I CLa,kD;MADrC,e;MACsC,0B;MAAyB,gB;MAD/D,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAEI,qEAGW,CA HX,EAGc,IAHd,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA, iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,+EAGgB,CAHhB,EAGmB,IAHnB,C;MAKA,yEAGa,CAHb,EAGgB,I AHhB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,6EAGe,CAHf,EAGkB,IAHIB,C;MAKA,6FAGuB,CA HvB,EAG0B,IAH1B,C;MAKA,yFAGqB,CAHrB,EAGwB,IAHxB,C;MAKA,4EAGc,EAHd,EAGkB,IAHIB,C;MA KA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,8EAGe,EAHf,EAGmB,I AHnB,C;MAKA,wFAGoB,EAHpB,EAGwB,IAHxB,C;MAKA,gEAGQ,EAHR,EAGY,IAHZ,C;MAKA,8DAGO,E AHP,EAGW,IAHX,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,oEAGU,EAHV,EAGc,IAHd,C;MAKA,k FAGiB,EAHjB,EAGqB,IAHrB,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,gFAGgB,EAHhB,EAGoB,I AHpB,C;MAKA,4FAGsB,EAHtB,EAG0B,IAH1B,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,wEAGY, EAHZ,EAGgB,IAHhB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB, C;MAKA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,oGAG0B,EAH1B,EAG8B,IAH9B,C;MAKA,gGAGwB,EAHxB ,EAG4B,IAH5B,C;MAUA,oC;K;;IA3JA,+C;MAAA,yB;MAAA,uC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA ,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD; MAAA,yB;MAAA,yC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,2D;M AAA,yB;MAAA,mD;K;;IAKA,yD;MAAA,yB;MAAA,iD;K;;IAKA,kD;MAAA,yB;MAAA,0C;K;;IAKA,iD;MAA A,yB;MAAA,yC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,wD;MAAA, yB;MAAA,gD;K;;IAKA,4C;MAAA,yB;MAAA,oC;K;;IAKA,2C;MAAA,yB;MAAA,mC;K;;IAKA,gD;MAAA,yB; MAAA,wC;K;;IAKA,8C;MAAA,yB;MAAA,sC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,sD;MAAA,yB;MA AA,8C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,0D;MAAA,yB;MAAA,kD;K;;IAKA,sD;MAAA,yB;MAAA, 8C;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,oD;MAAA,yB;MAAA,4C; K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,8D;MAAA,yB;MAAA,sD;K;;IAKA,4D;MAAA,yB;MAAA,oD;K;8 CAKA,gB;MAG2D,OAAK,iBAAL,IAAK,CAAL,KAA2B,IAAK,c;K;IAE3F,kC;MAAA,sC;K;uDACI,oB;MAEQ,I ADE,QACF,IAAG,CAAH,IADE,QACF,IAAM,EAAN,C;QADJ,OACgB,sBAAS,QAAT,C;WACZ,IAFE,QAEF,IA AG,EAH,IAFE,QAEF,IAAO,EAAP,C;QAFJ,OAEiB,sBAAS,WAAW,CAAX,IAAT,C;;QACL,MAAM,gCAAYB ,eAAY,QAAZ,qBAAZB,C;K;;;IAL1B,8C;MAAA,yB;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7JJ,+B;MAAA,+yC ;K;;IAAA,oC;MAAA,a;AAAA,Y;UAAA,4C;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aA AA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,kB;UAAA,kD;aAAA,gB;UAAA,gD;aAAA,wB;UAAA,wD;aAAA,sB ;UAAA,sD;aAAA,e;UAAA,+C;aAAA,c;UAAA,8C;aAAA,iB;UAAA,iD;aAAA,gB;UAAA,gD;aAAA,qB;UAAA,q D;aAAA,S;UAAA,yC;aAAA,Q;UAAA,wC;aAAA,a;UAAA,6C;aAAA,W;UAAA,2C;aAAA,kB;UAAA,kD;aAAA, mB;UAAA,mD;aAAA,iB;UAAA,iD;aAAA,uB;UAAA,uD;aAAA,mB;UAAA,mD;aAAA,a;UAAA,6C;aAAA,iB;U AAA,iD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,2B;UAAA,2D;aAAA,yB;UAAA,yD;;UAAA,6D;;K;;ICKi D,2C;uBAA+B,O;;K;;IAC5E,8C;MAAA,kE;MAAuB,qCAAK,IAAL,C;MAAvB,Y;K;ICD8B,gC;MAe9B,gBAAiC ,YAAy,SAAhB,GAA2B,OAA3B,GAAwC,E;K;uFAGjE,Y;MAAQ,OAAO,aAAY,O;K;yCAE/B,iB;MACW,gBAA P,a;MrGoGG,Q;MAAA,IqGpGc,KrGoGV,IAAS,CAAT,IqGpGU,KrGoGI,IAAS,2BAA3B,C;QAAA,OAAc,qBq GpGxB,KrGoGwB,C;;QqGpGf,MAAM,8BAA0B,mCAAYB,WAAzB,MAA1B,C;;MAAhC,W;K;kDAEJ,gC;MAA gF,OAAA,a1GiMY,W0GjMK,U1GiML,E0GjMiB,Q1GiMjB,C;K;6C0G/L5F,iB;MACI,qCAAU,KAAV,C;MACA, OAAO,I;K;6CAGX,iB;MACI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,uC;MACI,OAAA,IAAK,q BAAY,wBAAS,MAArB,EAA6B,UAA7B,EAAyC,QAAzC,C;K;sCAET,Y;MAayB,UAEK,M;MAL1B,eAAe,E;M ACf,YAAy,aAAO,OAAP,GAAgB,CAAhB,I;MACZ,OAAO,SAAS,CAAhB,C;QACI,UAAU,0BAAO,YAAP,EA A O,oBAAP,Q;QACV,IAAQ,eAAJ,GAAL,CAAJ,IAAwB,SAAS,CAArC,C;UACI,WAAW,0BAAO,cAAP,EAAsB AAP,U;UACX,IAAS,gBAAL,IAAK,CAAT,C;YACI,WAAW,+BAAW,iBAAX,wBAakB,gBAAlB,C;;YAEX,WA AW,+BAAW,gBAAX,wBAAiB,iBAAjB,C;;;UAGf,gCAAY,GAAZ,C;;;MAGR,gBAAS,Q;MACT,OAAO,I;K;6CA GX,iB;MAOI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAQI,iBAAU,K;MACV,OAAO,I;K;6C AGX,iB;MAQI,iBAAGB,eAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAC2C,2BAAO,KAAP,C;K;6CAE3C ,iB;MAOI,gBAAA,IAAK,SAAL,IAAe,wBAAS,MAAxB,C;MACA,OAAO,I;K;uCAGX,Y;MAU6B,kB;K;qDAE7



B,2B;K;8CAcA,kB;MAO0C,OAAA,IAAY,SAAY,SAAQ,MAAR,C;K;8CAEIE,8B;MAQ2D,OAAA,IAAY,SAAY,  
SAAQ,MAAR,EAAGB,UAAhB,C;K;kDAEnF,kB;MAQ8C,OAAA,IAAY,SAAY,aAAY,MAAZ,C;K;kDAEIE,8B;  
MASI,IAAI,MpGuGwC,YAAU,CoGvGID,IAAoB,aAAa,CAArC,C;QAAwC,OAAO,E;MAC/C,OAAO,IAAY,SA  
AY,aAAY,MAAZ,EAAoB,UAApB,C;K;4CAGnC,wB;MAWI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MA  
Eb,gBAAS,a1GmB+E,W0GnB9D,C1GmB8D,E0GnB3D,K1GmB2D,C0GnB/E,YAA6B,KAA7B,IAAqC,a1GgB2  
B,W0GhBV,K1GgBU,C;M0GfzE,OAAO,I;K;6CAGX,wB;MAQI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;  
MAEb,gBAAS,a1GK+E,W0GL9D,C1GK8D,E0GL3D,K1GK2D,C0GL/E,uBAA6B,kBAA7B,IAAqC,a1GE2B,W0  
GFV,K1GEU,C;M0GDzE,OAAO,I;K;6CAGX,wB;MAUI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,g  
BAAS,a1GX+E,W0GW9D,C1GX8D,E0GW3D,K1GX2D,C0GW/E,GAAmC,eAAN,KAAM,CAAnC,GAAsD,a1G  
dU,W0GcO,K1GdP,C;M0GezE,OAAO,I;K;6CAGX,wB;MAA1,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MA  
Eb,gBAAS,a1G9B+E,W0G8B9D,C1G9B8D,E0G8B3D,K1G9B2D,C0G8B/E,GAAmC,SAAN,KAAM,CAAnC,GA  
AgD,a1GjCgB,W0GiCC,K1GjCD,C;M0GkCzE,OAAO,I;K;6CAGX,wB;MAWI,oCAAA,4BAAmB,KAAAnB,EAA0  
B,WAA1B,C;MAEb,gBAAS,a1G/C+E,W0G+C9D,C1G/C8D,E0G+C3D,K1G/C2D,C0G+C/E,GAAmC,SAAN,KA  
AM,CAAnC,GAAGD,a1GldGB,W0GkDC,K1GIDD,C;M0GmDzE,OAAO,I;K;6CAGX,wB;MACuD,2BAAO,KAA  
P,EAAC,KAAc,C;K;6CAEvD,wB;MAUI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,eAAe,wBAAS,M;  
MACxB,gBAAC,IAAK,S1GnEqE,W0GmEpD,C1GnEoD,E0GmEjD,K1GnEiD,C0GmE1E,GAAC,C,QAA1C,GAA6  
C,IAAK,S1GtES,W0GsEQ,K1GtER,C;M0GuEzE,OAAO,I;K;gDAGX,qB;MAcI,IAAI,YAAY,CAAhB,C;QACI,M  
AAM,gCAAYB,0BAAuB,SAAvB,MAAzB,C;MAGV,IAAI,aAAa,WAAjB,C;QACI,gBAAS,a1G1F2E,W0G0F1D,  
C1G1F0D,E0G0FvD,S1G1FuD,C;Q0G4FpF,aAAU,WAAV,MAAuB,SAAvB,M;UACI,qCAAU,CAAV,C;K;gD  
AKZ,sB;MAQI,oCAAA,4BAAmB,UAAAnB,EAA+B,WAA/B,C;MAEb,OAAO,a1G/GkE,W0G+GjD,U1G/GiD,C;K;  
gD0GkH7E,gC;MAQI,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,WAAzC,C;MAEb,OAAO,a1GzHiF,W  
0GyHhE,U1GzHgE,E0GyHpD,Q1GzHoD,C;K;yC0G4H5F,Y;K;uCACa,Y;MAAKC,oB;K;oCAEIC,Y;MAOI,gBA  
AS,E;MACT,OAAO,I;K;0CAGX,wB;MAQI,oCAAA,2BAAkB,KAAIB,EAAyB,WAAzB,C;MAEb,gBAAS,a1GjK  
+E,W0GiK9D,C1GjK8D,E0GiK3D,K1GjK2D,C0GiK/E,uBAA6B,kBAA7B,IAAqC,a1GpK2B,W0GoKV,QAAQ,  
CAAR,I1GpKU,C;K;+C0GuK7E,uC;MAYI,yBAAkB,UAAIB,EAA8B,QAA9B,EAAwC,WAAxC,C;MAEA,gBA  
Ac,IAAK,S1GILqE,W0GkLpD,C1GILoD,E0GkLjD,U1GILiD,C0GkL1E,GAAuC,KAAvC,GAA+C,IAAK,S1GrL  
O,W0GqLU,Q1GrLV,C;M0GsLzE,OAAO,I;K;kDAGX,wC;MACI,IAAI,aAAa,CAAb,IAAkB,aAAa,MAAnC,C;Q  
ACI,MAAM,8BAA0B,iBAAc,UAAAd,kBAAMC,MAA7D,C;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAAY  
B,gBAAa,UAAb,qBAAqC,QAArC,MAAzB,C;K;+CAId,iB;MAYI,oCAAA,2BAAkB,KAAIB,EAAyB,WAAzB,C;  
MAEb,gBAAS,a1G7M+E,W0G6M9D,C1G7M8D,E0G6M3D,K1G7M2D,C0G6M/E,GAA6B,a1GhNmC,W0GgNI  
B,QAAQ,CAAR,I1GhNkB,C;M0GiNzE,OAAO,I;K;kDAGX,gC;MAWI,yBAAkB,UAAIB,EAA8B,QAA9B,EAA  
wC,WAAxC,C;MAEA,gBAAS,a1G9N+E,W0G8N9D,C1G9N8D,E0G8N3D,U1G9N2D,C0G8N/E,GAAK,C,a1GjO  
8B,W0GiOb,Q1GjOa,C;M0GkOzE,OAAO,I;K;kDAGX,gE;MAc+C,iC;QAAA,oBAAYB,C;MAAG,0B;QAAA,aA  
AkB,C;MAAG,wB;QAAA,WAAgB,IAAK,O;MAKIF,IACf,I;MALhB,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B  
,EAAyC,WAAzC,C;MACb,oCAAA,4BAAmB,iBAAnB,EAAc,oBAAoB,QAApB,GAA+B,UAA/B,IAAtC,EAAiF  
,WAAy,OAA7F,C;MAEb,eAAe,iB;MACf,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,YAAY,eAAZ,EAAy,uBAAZ,  
UAA0B,yBAAO,KAAP,C;K;kDAIIC,uC;MAcI,iBAAGB,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,C;MACH  
B,OAAO,I;K;kDAGX,uC;MAYI,gBAAgB,KAAM,W;MACTB,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAy  
C,SAAU,OAAAnD,C;MAEb,iBAAU,S1G3R8E,W0G2R1D,U1G3R0D,E0G2R9C,Q1G3R8C,C;M0G4RxF,OAAO,I;  
K;kDAGX,8C;MAGBI,oCAAA,4BAAmB,KAAAnB,EAA0B,IAAK,OAA/B,C;MAEb,gBAAS,a1GjT+E,W0GiT9D,C  
1GjT8D,E0GiT3D,K1GjT2D,C0GiT/E,GAAmC,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,CAAnC,GAA0E,a  
1GpTV,W0GoT2B,K1GpT3B,C;M0GqTzE,OAAO,I;K;kDAGX,8C;MAGBI,oCAAA,4BAAmB,KAAAnB,EAA0B,W  
AA1B,C;MAEb,gBAAgB,KAAM,W;MACTB,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,SAAU,OAAAnD  
,C;MAEb,gBAAS,a1G1U+E,W0G0U9D,C1G1U8D,E0G0U3D,K1G1U2D,C0G0U/E,GAA6B,S1G1UkD,W0G0U9  
B,U1G1U8B,E0G0UIB,Q1G1UkB,C0G0U/E,GAAyE,a1G7UT,W0G6U0B,K1G7U1B,C;M0G8UzE,OAAO,I;K;I  
AliBX,6C;MAAA,uD;MAKOC,2B;MALpC,Y;K;IAQA,8C;MAAA,uD;MAC4C,0BAAK,OAAQ,WAAb,C;MAD5  
C,Y;K;IAGA,qC;MAAA,uD;MACuB,0BAAK,EAAL,C;MADvB,Y;K;2EA4hBJ,qB;MAOG,E,OAAA,SAAK,Q;K;u  
EAERe,mC;MAQ+E,SAAK,aAAI,KAAJ,EAAW,KAAX,C;K;+EAEPf,kD;MAAI,OAAA,SAAK,kBAAS,UAAAT,E

AAqB,QAARb,EAA+B,KAA/B,C;K;+EAET,4B;MAY6E,OAAA,SAAK,kBAAS,KAAT,C;K;qFAEIF,2C;MAWo  
G,OAAA,SAAK,qBAAY,UAAZ,EAawB,QAAxB,C;K;uFAEZG,2E;MAe2E,iC;QAAA,oBAAYB,C;MAAG,0B;Q  
AAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC7I,SAAK,qBAAY,WAAZ,EAAYB,iBAAzB,EAA4C,  
UAA5C,EAawD,QAAxD,C;K;qFAET,kD;MAeI,OAAA,SAAK,qBAAY,KAAZ,EAAMb,UAAAnB,EAA+B,QAA/  
B,C;K;uFAET,kD;MAaI,OAAA,SAAK,qBAAY,KAAZ,EAAMb,UAAAnB,EAA+B,QAA/B,C;K;qFAET,yD;MAiBI  
,OAAA,SAAK,qBAAY,KAAZ,EAAMb,KAAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;uFAET,yD;MAiBI,OAA  
A,SAAK,qBAAY,KAAZ,EAAMb,KAAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;qF3GhsBT,qB;MAMoD,OA6B  
W,8BAAY,cAfrB,YAAY,CAAZ,C;K;yFAZtD,qB;MAYsD,OAeS,8BAAY,cAfrB,YAAY,CAAZ,C;K;iFAEtD,qB;  
MAaoD,OAAW,8BAAY,c;K;qFAE3E,yB;MAAA,uD;MAAA,4B;QAMoD,+B;O;KANpD,C;IAQA,kC;MAYI,gB  
AiB2D,8BAAY,c;MAhBvE,OAAW,SAAU,OAAV,GAAMb,CAAvB,GAA0B,SAA1B,GAAoC,qBAAU,CAAV,C;  
K;iFAG/C,qB;MAaoD,OAAW,8BAAY,c;K;IAE3E,kC;MAU+C,mC;K;IAE/C,oC;MAGoD,QAAQ,cAAA,sCAAK,  
mBAAL,EAAYB,sCAAK,mBAA9B,CAAR,6B;K;IAEpD,mC;MAGmD,QAAQ,cAAA,sCAAK,kBAAL,EAawB,s  
CAAK,kBAA7B,CAAR,6B;K;IAO/C,iC;MAAQ,OAAA,oCAAA,iBAAQ,2BAAR,C;K;IAEzB,8B;MAOI,IAAI,YA  
AO,GAAX,C;QACI,OAAO,I;MAEX,OAAO,gCAA8C,mD;K;IAGzD,6B;MAUI,IAAI,CAAQ,kBAAK,GAAL,CA  
AR,iCAAoB,CAAQ,kBAAK,EAAL,CAAR,6BAAXB,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OA  
AO,K;MAEX,OAAO,uB;K;IAGX,oC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ,kBAAK,EAAL  
L,CAAR,6BAAPB,IAAwC,CAAQ,kBAAK,EAAL,CAAR,6BAA5C,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX  
C;QACI,OAAO,K;MAGX,OAAO,0BAAiB,uB;K;IAG5B,4B;MASI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAA  
J,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAEX,OAAO,sB;K;IAGX,gC;MAUI,IAAI,  
CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAEX,  
OAAO,0B;K;IAGX,gC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAA  
O,GAAX,C;QACI,OAAO,K;MAEX,OAAO,0B;K;IAGX,gC;MASI,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAE  
X,OAAO,gCAAoD,yD;K;IAG/D,iC;MAUI,OAAO,aAAQ,EAAR,IAAoB,CAAQ,mBAAU,GAAV,CAAR,6B;K;IA  
G/B,iC;MAMiD,kC;K;iF4GtPjD,yB;MAAA,+C;MAAA,4B;QAMuD,OAAK,UAAAL,SAAK,C;O;KAN5D,C;IAQA,  
gC;MAMiD,4B;MAAA,S;QAAGB,cAAA,S3G4LC,c2G5LD,EAaoB,MAApB,C;MAAhB,W;K;IAEjD,6B;MAI0C  
,Q;MAAA,yDAaKB,kBAaKB,SAAIB,C;K;IAE5D,oC;MAK0D,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SAA  
IB,C;K;IAG3E,8B;MAI4C,Q;MAAA,0DAAMb,kBAaKB,SAAIB,C;K;IAE/D,qC;MAKsD,Q;MAAA,0CAAc,KAA  
d,oBAawB,kBAaKB,SAAIB,C;K;IAE9E,0B;MAIwC,Q;MAAA,wDAAiB,kBAaKB,SAAIB,C;K;IAEzD,mC;MA  
KkD,Q;MAAA,wCAAY,KAAZ,oBAAsB,kBAaKB,SAAIB,C;K;IAExE,2B;MAI0C,Q;MAAA,yDAaKB,kBAaKB,  
SAAIB,C;K;IAE5D,oC;MAK0D,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SAAIB,C;K;IAE3E,6B;MAIyF,kB  
AA1C,CAAO,S;MACID,IAAO,QrHeD,WqHfC,CAAH,IAAc,CAAM,kBAAPB,KrHeE,WqHf6B,KAAM,GAAN,I  
AAkB,kBAAjD,CAAJ,C;QACI,4B;MAFsC,OrHiBnC,W;K;6EqHZX,yB;MAAA,6C;MAAA,4B;QAKmD,0B;O;K  
ALnD,C;IAOA,mC;MAIgG,kBAA1C,CAAO,S;MAAR,OACjD,EAAK,QrH2BgB,WqH3BhB,CAAH,IAAc,CAA  
M,kBAAPB,KrH2BmB,WqH3BY,KAAM,GAAN,IAAkB,kBAAjD,CAAF,Ch2BO,GAAqB,WAArB,GAA+B,I;K  
;yFqHxB1C,yB;MAAA,yD;MAAA,4B;QAK0D,gC;O;KAL1D,C;iFAOA,yB;MAAA,6C;MAAA,mC;QAO6D,OA  
Aa,SAAR,SAAQ,EAAS,KAAT,C;O;KAP1E,C;iFASA,yB;MAAA,6C;MAAA,mC;QAO8D,OAAa,SAAR,SAAQ,E  
AAS,KAAT,C;O;KAP3E,C;IASA,sC;MAMqD,OAAA,SAAY,UAAS,WAAW,KAAX,CAAT,C;K;IAEjE,4B;MAA  
sC,QAAM,S3G4EsB,c2G5E5B,C;aACIC,K;aAAA,M;aAAA,M;UADkC,OACT,I;UADS,OAE1B,K;K;IAGZ,2B;  
MAKI,IAAI,EAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYB,WAAQ,KAAR,kCAAzB,C;MAEV,  
OAAO,K;K;IAGX,8B;MAA2D,Q;MACvD,YAAQ,EAAR,IAAe,QAAQ,EAavB,C;QAA8B,cAAO,E;WACrC,YA  
AQ,EAAR,IAAe,QAAQ,EAavB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,YAAQ,EAAR,IAAe,QAAQ,G  
AAvB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,WAAO,GAAP,C;QAAMb,S;WACnB,YAAQ,KAAR,IA  
AoB,QAAQ,KAA5B,C;QAAwC,cAAO,KAAP,GAaKB,EAAlB,I;WACxC,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,  
C;QAAwC,cAAO,KAAP,GAaKB,EAAlB,I;QAC3B,sBAAL,IAAK,C;MrH9CN,a;MqHuCgD,OAQ/C,WAAJ,GA  
AiB,EAajB,GAAyB,E;K;ICIJG,2C;MAHpC,e;MAGqC,kB;MAHrC,iB;MAAA,uB;K;IAAA,kC;MAAA,qC;O;MA  
II,qEACY,GADZ,C;MAEA,iEAIU,GAJV,C;K;IAFA,+C;MAAA,wB;MAAA,uC;K;IAEA,6C;MAAA,wB;MAAA  
,qC;K;IANJ,8B;MAAA,mF;K;IAAA,mC;MAAA,a;aAAA,a;UAAA,4C;aAAA,W;UAAA,0C;UAAA,4D;K;IAa  
wG,4B;MAAE,OAAA,EAAG,M;K;IAA7G,qC;MAAQE,iCAAA,EAAb,EAA0B,OAA1B,0BAAMc,cAANc,C;K;IA

QIC,2B;MAAC,kB;K;;sCALpC,Y;MAKoC,iB;K;wCALpC,iB;MAAA,sBAKoC,qCALpC,C;K;oCAAA,Y;MAAA,  
OAKoC,iDALpC,M;K;oCAAA,Y;MAAA,c;MAKoC,sD;MALpC,a;K;kCAAA,iB;MAAA,2IAKoC,sCALpC,G;K;I  
AqB0B,iC;MA8PtB,6B;MArPA,eACoC,O;MACpC,eACsD,QAAR,OAAQ,C;MACtD,uBAAoC,WAAO,OAAP,E  
AAwB,QAAR,OAAQ,EAAQ,IAAR,CAAxB,C;MACpC,6BAA2C,I;MAI3C,oCAAKd,I;K;0CAHID,Y;MACI,Q;M  
AAA,U;MAAA,gD;QAAA,a;;QAA8D,gBAAvC,WAAO,YAAP,EAAwB,QAAR,YAAQ,EAAQ,IAAR,CAAxB,C;  
QAA8C,6BtHmCnE,S;QsHnCF,StHoCG,S;;MsHpCH,a;K;iDAGJ,Y;MACI,Q;MAAA,U;MAAA,uD;QAAA,a;;QtH  
VG,gB;QsHWC,IAAY,aAAR,YAAQ,EAAW,EAAX,CAAR,IAAmC,WAAR,YAAQ,EAAS,EAAT,CAAvC,C;UA  
AA,eACI,oB;;UAEA,OAAO,WAAO,MAA2B,UAAf,YAAR,YAAQ,qBAAU,EAAV,EAAe,qBAAQ,EAAR,EAA3  
B,MAAP,EAA2D,QAAR,YAAQ,EAAQ,IAAR,CAA3D,C;QACb,4B;QAAO,oCtH0BP,S;QsH/BF,StHgCG,S;;Ms  
HhCH,a;K;sCAQJ,iB;MAEkB,MAAd,oBAAC,C;MACd,YAAY,oBAAC,MAAK,KAAM,WAAX,C;MAC1B,OAA  
O,iBAAiB,KAAM,MAAN,KAAe,CAAhC,IAAqC,oBAAC,UAAf,KAA2B,KAAM,O;K;8CAGjF,iB;MAEkB,MAA  
d,oBAAC,C;MACd,OAAO,oBAAC,MAAK,KAAM,WAAX,C;K;wCAGzB,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,  
QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,cAAc,  
0B;MACd,oBAAoB,K;MACpB,OAAO,OAAQ,MAAK,KAAM,WAAX,C;K;mCAGnB,6B;MAS4C,0B;QAAA,aA  
AkB,C;MAC1D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,MAAM,8BAA0B,gCAA6B,UAA7B,w  
BAAyD,KAAM,OAAzF,C;;MAEV,OAAqB,SAAd,oBAAC,EAAS,KAAM,WAAf,EAA2B,UAA3B,EAAuC,oBAA  
vC,C;K;IAeG,6E;MAAA,mB;QAAE,+BAAK,aAAL,EAAY,kBAAZ,C;O;K;IAA2B,uC;MAAW,OAAA,KAAM,O;  
K;sCAZ1E,6B;MAQ+C,0B;QAAA,aAAkB,C;MAC7D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,  
MAAM,8BAA0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAO,mBAAiB,6CAAjB,EAA8C,sB  
AA9C,C;K;0CAGX,iB;MAMI,OAA2B,SAA3B,iCAA2B,EAAS,KAAM,WAAf,EAA2B,CAA3B,EAA8B,oBAA9  
B,C;K;sCAE/B,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB  
,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,OAA2B,SAApB,0BAAoB,EAAS,KAAM,WAAf,EAA2B,KAA3B,  
EAAkC,oBAAIC,C;K;IA4BL,mD;MAAA,qB;QAAE,2BAAoB,EAAPB,EAAwB,mBAAxB,C;O;K;sCAxB5B,8B;  
MAqBI,IAAI,CAAa,YAAZ,WAAy,EAAS,EAAT,CAAb,IAA+B,CAAa,YAAZ,WAAy,EAAS,EAAT,CAAhD,C;  
QACI,OAAO,KAAM,W5G2E4E,S4G3EnD,oB5G2EmD,E4G3EpC,W5G2EoC,C;;M4GzE7F,OAAO,qBAAQ,KA  
AR,EAAe,iCAAF,C;K;sCAGX,4B;MAMI,YAAY,kBAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAmB,OAAO,KAA  
M,W;MAEHc,gBAAgB,C;MACHb,aAAa,KAAM,O;MACnB,SAAS,mBAAc,MAAd,C;;QAEI,iBAAiB,oB;QACj  
B,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAYB,UAAW,MAAM,MAA1C,C;QACH,EAAG,gBAAO,UAAU,UAAV  
,CAAP,C;QACH,YAAY,UAAW,MAAM,aAAjB,GAAgC,CAAhC,I;QACZ,QAAQ,UAAW,O;;MACd,oBAAy,M  
AAZ,IAAsB,aAAATB,C;MAET,IAAI,YAAY,MAAhB,C;QACI,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAYB,MAAZ  
B,C;;MAGP,OAAO,EAAG,W;K;2CAGd,8B;MA0BgB,Q;MALZ,IAAI,CAAa,YAAZ,WAAy,EAAS,EAAT,CAAb  
,IAA+B,CAAa,YAAZ,WAAy,EAAS,EAAT,CAAhD,C;QACI,uBAA+B,QAAR,YAAQ,EAAQ,GAAR,C;QAC/B,  
OAAO,KAAM,W5GoB4E,S4GpBnD,WAAO,YAAP,EAAGB,gBAAhB,C5GoBmD,E4GpBhB,W5GoBgB,C;;M4G  
jBjF,yBAAK,KAAL,C;MAAA,iB;QAAe,OAAO,KAAM,W;;MAAxC,YAAY,I;MCoLO,gBAAhB,sB;MDjLC,yBt  
G2LgF,0BsG3LzD,CtG2LyD,EsG3LhD,WAAM,MtG2L0C,CAAKC,WsG3LIH,C;MACA,yBAAO,uCAAP,C;MA  
CA,yBtGyLgF,0BsGzLnD,WAAM,KAAZ,GAAMB,CAAnB,ItGyLyD,EsGzL7B,YtGyL6B,CAAKC,WsGzLIH,C;  
MAHJ,OtHIJG,SuHoUqC,W;K;oCD3K5C,wB;MAO6C,qB;QAAA,QAAa,C;MAMxC,Q;MALd,wBAAwB,KAAx  
B,C;MtHrIG,SsHsIW,qBAAQ,KAAR,C;MAAd,cAAuC,UAAS,CAAb,GAAgB,EAhB,GAA2B,OAAH,EAAG,E  
AAK,QAAQ,CAAR,IAAL,C;MAC9D,ajI3jGd,gB;MiI4JhD,gBAAgB,C;MAEF,yB;MAAd,OAAc,cAAAd,C;QAAc,  
uB;QACV,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,MAAM,MAAnC,CAA0C,WAApD,C;  
QACP,YAAY,KAAM,MAAM,aAAZ,GAA2B,CAA3B,I;;MAEHb,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ  
,EAAuB,KAAM,OAA7B,CAAqC,WAA/C,C;MACP,OAAO,M;K;IAgBS,yI;MAAA,wC;MAAA,6B;MAAA,yB;M  
AAA,0C;MAAA,oC;MAAA,0C;MAAA,yB;MAAA,6B;MAAA,8B;MAAA,8B;MAAA,kC;K;;;gEAAA,Y;;;iCA  
CA,mCAAK,wBAAL,C;cACZ,IAAI,4BAAiB,6BAAS,CAA9B,C;gBACI,gB;gCAAA,iCAAM,wBAAM,WAAZ,O  
;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,gB;;;;;cAEI,M;;qCAGY,C;sCACC,C;cAEjB,gB;;;sCACqB,+B;cACj  
B,gB;8BAAA,iCtGuI4E,mBsGvItE,wBtGuIsE,EsGvItD,oBtGuIsD,EsGvI3C,qBAAW,MAAM,MtGuI0B,CAAKC,  
WsGvI9G,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,uBAAy,qBAAW,MAAM,aAAjB,GAAgC,CAAhC,I;cACZ,  
mBAAQ,qBAAW,O;cAJvB,KAKS,qDALT,EAKS,qBALT,OAKyB,2BAAQ,CAAR,IALzB,KAKsC,gBALtC,S;gB

AAA,gB;;;cAAA,gB;;;cAOA,gB;8BAAA,iCtGkIgF,mBsGIIIE,wBtGkIOE,EsGIIID,oBtGkIOE,EsGII/C,wBAAM,  
OtGkIyC,CAAKC,WsGIIH,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAhBA,OAgBA,a;,,,,,;K;IAjBY,sF;MAAA,  
yD;uBAAA,6H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;8CAbpB,wB;MAUuD,qB;QAAA,QAAa,C;MACHe,wBAA  
wB,KAAxB,C;MAEA,OAAO,SAAS,gDAAT,C;K;+BAxBX,Y;MAMyC,OAAA,oBAAc,W;K;IAEvD,2B;MAAA,+  
B;MAMBI,uBAA4B,WAAO,uBAAP,EAAiC,GAAjC,C;MAC5B,2BAAgC,WAAO,SAAP,EAAoB,GAApB,C;MA  
GhC,iCAAsC,WAAO,KAAP,EAAiB,GAAjB,C;K;oDatBtC,mB;MAIwD,oBAAM,oBAAO,OAAP,CAAN,C;K;+C  
AExD,mB;MAIoD,OAAA,O5GnEyC,S4GmEnB,oB5GnEmB,E4GmEJ,M5GnEI,C;K;0D4GqE7F,mB;MAI+D,OA  
AA,O5GzE8B,S4GyER,wB5GzEQ,E4GyEW,M5GzEX,C;K;gE4G8E7F,mB;MAAgE,OAAA,O5G9E6B,S4G8EP,8  
B5G9EO,E4G8EkB,M5G9EIB,C;K;;I4GwDjG,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;;IA5PA,4C;MAAA,+C;M  
ACkE,kBAAK,OAAAL,EAAc,MAAM,MAAN,CAAd,C;MADIE,Y;K;IAGA,sC;MAAA,+C;MAC6C,kBAAK,OAA  
L,EAAc,UAAAd,C;MAD7C,Y;K;IA4RO,kG;MAAA,kC;MAAA,8C;MAAA,kC;MAAA,kC;MACH,uBAA+B,a;MA  
I/B,sF;MAOA,sBAA0C,I;K;+FAX1C,Y;MAAA,2B;K;+FAEI,Y;MAAQ,qBAAA,kBN/R8C,CM+RxC,CN/RwC,C  
M+R9C,C;K;gGAEZ,Y;MAAA,4B;K;IAY2B,oG;MAAA,kC;MAAS,uB;K;mJACG,Y;MAAQ,OAAA,kBAAM,O;  
K;wGACrC,iB;MAAuC,Q;MAAA,eAAA,kBN/SG,CM+SG,KN/SH,CM+SH,mBAAGB,E;K;;qGAJnE,Y;MACI,IA  
AI,2BAAJ,C;QACI,yH;;MAKJ,OAAO,kC;K;4CAGf,Y;MACI,OAAy,SAAZ,wBAAy,EAAS,kBAAT,EAAoB,kB  
AAM,UAAV,GAAqB,8BAAuB,kBAAM,MAA7B,CAArB,GAA8D,kBAAM,aAAN,GAAqB,CAArB,IAA9E,EAA  
sG,wBAATG,C;K;gEAEhB,iB;MACI,IAAI,QAAc,iBAAN,kBAAM,CAAIB,C;QACI,YAAkB,kBAAy,YAAW,KA  
AX,C;QAC9B,IAAa,KAAT,sBAAiB,KAAR,B,C;UACI,YAAkB,kBAAy,YAAW,QAAQ,CAAR,IAAX,C;UAC9B,I  
AAa,KAAT,sBAAiB,KAAR,B,C;YACI,OAAO,QAAQ,CAAR,I;K;MAInB,OAAO,QAAQ,CAAR,I;K;IAjCiC,oE;M  
AAA,kC;MAA+B,6B;K;mHACHD,Y;MAAQ,OAAA,kBAAM,O;K;IACqC,4E;MAAA,qB;QAAE,yBAAK,EAAL,  
C;O;K;qEAA5E,Y;MAAiD,OAAqB,OAAb,aAAR,oBAAQ,CAAa,EAAL,iEAAJ,CAAiB,W;K;wEACvF,iB;MAA4  
C,Q;MAAA,eAAA,kBNpSU,CMoSJ,KNpSI,CMoSv,YAAoB,oBAApB,O;K;;IAdxD,uD;MACI,sBAAiB,I;MACj  
B,YAAy,eAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAMB,OAAO,I;MAC1B,YAAy,aAAA,KAAM,MAAN,EAAa,  
sBAAy,CAAZ,IAAb,C;MAEZ,mE;K;IA2CJ,iD;MAM+B,UAKO,M;MATIC,YAAy,C;MACZ,aAAa,mBAAc,WA  
AY,OAA1B,C;MAEb,OAAO,QAAQ,WAAy,OAA3B,C;QACI,WAAW,wBAAy,YAAZ,EAAY,oBAAZ,Q;QACX  
,IAAI,SAAQ,EAZ,C;UACI,IAAI,UAAS,WAAy,OAAzB,C;YACI,MAAM,gCAAyB,mCAAZB,C;UAEV,MAAO  
,gBAAO,wBAAy,cAAZ,EAAY,sBAAZ,UAAP,C;eACJ,IAAI,SAAQ,EAZ,C;UACH,IAAI,UAAS,WAAy,OAAz  
B,C;YACI,MAAM,gCAAyB,kCAAZB,C;UAEV,IAAI,uBAAy,KAAY,MAAsB,GAA1B,C;YACI,MAAM,gCAAy  
B,4DAAzB,C;UAEV,IAAI,EAuB,kBAAK,EAAL,CAAvB,0CAAy,KAAY,EAAL,C;YACI,MAAM,gCAAyB,mC  
AAzB,C;UAEV,eAA2B,eAAZ,WAAy,EAae,KAaf,EAAsB,KAAM,YAAy,KAAxC,C;UAC3B,iBAAWD,MAAv  
C,W5G7KmE,W4G6K7C,K5G7K6C,E4G6KtC,Q5G7KsC,C4G6K5B,C;UAExD,IAAI,cAAc,KAAM,YAAy,KAA  
pC,C;YACI,MAAM,8BAA0B,sBAAmB,UAAAnB,oBAA1B,C;UAEV,MAAO,gBAAO,KAAM,YAAN,aAAkB,UA  
AIB,CAAP,C;UACP,QAAQ,Q;;UAER,MAAO,gBAAO,IAAP,C;;MAGf,OAAO,MAAO,W;K;IAGIB,2D;MAEI,Y  
AAy,aAAa,CAAb,I;MACZ,iBAAiB,qBAAK,UAAL,IAAmB,E;MAGpC,OAAO,QAAQ,gBAAR,IAAkB,CAAe,k  
BAAK,EAAL,CAAF,wCAAK,KAAL,EAzB,C;QACI,oBAAoB,CAAC,aAAa,EAAb,IAAD,KAAqB,qBAAK,KA  
AL,IAAc,EAAnC,K;QACpB,IAAqB,CAAjB,qCAAyB,UAA7B,C;UACI,aAAa,a;UACb,qB;;UAEA,K;;MAGR,O  
AAO,K;K;I5GraX,yB;MAQiB,Q;MADb,aAAa,E;MACb,wBAAa,KAAb,gB;QAAa,WAAb,UAAa,KAAb,O;QACI,  
8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,yC;MAA+B,Q;MAH3B,IAAI,SAAS,CAAT,IAAc,SAAS,CAAvB,IA  
A4B,CAAA,KAAM,OAAN,GAAa,MAAb,QAAsB,MAATD,C;QACI,MAAM,8BAA0B,WAAS,KAAM,OAAf,kB  
AA+B,MAA/B,kBAAgD,MAA1E,C;MACV,aAAa,E;MACc,gBAAS,MAAT,I;MAA3B,iBAAc,MAAd,wB;QACI,  
8BAAU,MAAM,KAAN,CAAV,C;;MAEJ,OAAO,M;K;IAGX,mC;MAOiB,Q;MADb,aAAa,E;MACb,wBAAa,SA  
Ab,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,2D;MAY2C,0B;QAAA,a  
AAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MACjF,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAYC,SA  
AK,OAA9C,C;MACb,aAAa,E;MACb,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,8BAAU,UAAK,KAAL,CAAV,C;;M  
AEJ,OAAO,M;K;IASkB,gD;MAAA,qB;QAAE,+CAAI,EAAL,E;O;K;IAN/B,kC;MAMI,OAAO,kBAAU,gBAAV,  
EAAkB,+BAAIB,C;K;IAiBiC,oE;MAAA,qB;QAAE,+CAAI,qBAAa,EAAb,IAAJ,E;O;K;IAd9C,wD;MAYqC,0B;  
QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC3E,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EA  
AyC,gBAAzC,C;MACb,OAAO,kBAAU,WAAW,UAAx,IAAV,EAAL,C,2CAAjC,C;K;IAGX,mC;MAQI,OAAO,W

AAW,SAAX,EAAiB,CAAjB,EAAoB,gBAApB,EAA0B,KAA1B,C;K;IAGX,mF;MAeI,0B;QAAA,aAAkB,C;MAC  
IB,wB;QAAA,WAAGB,SAAK,O;MACrB,sC;QAAA,yBAaKc,K;MAEIC,oCAAA,4BAAMb,UAAAnB,EAA+B,QA  
A/B,EAAyC,SAAK,OAA9C,C;MACb,OAAO,WAAW,SAAX,EAAiB,UAAjB,EAA6B,QAA7B,EAAuC,sBAAvC,  
C;K;IAGX,sC;MAQI,OAAO,WAAW,SAAX,EAAiB,CAAjB,EAAoB,gBAApB,EAA4B,KAA5B,C;K;IAGX,sF;M  
AeI,0B;QAAA,aAAkB,C;MACIB,wB;QAAA,WAAGB,SAAK,O;MACrB,sC;QAAA,yBAaKc,K;MAEIC,oCAAA,  
4BAAMb,UAAAnB,EAA+B,QAA/B,EAAyC,gBAAZC,C;MACb,OAAO,WAAW,SAAX,EAAiB,UAAjB,EAA6B,Q  
AA7B,EAAuC,sBAAvC,C;K;uFAGX,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;  
uFAEIE,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;yFAEIE,qC;MACoF,OAAA,S  
AAY,SAAQ,GAAR,EAAa,SAAb,C;K;iGAehG,qC;MACwF,OAAA,SAAY,aAAY,GAAZ,EAAiB,SAAjB,C;K;+F  
AEpG,kC;MACiF,OAAA,SAAY,YAAW,CAAX,EAAc,QAAd,C;K;2FAE7F,wB;MACgE,OAAA,SAAY,UAA5,C  
AAT,C;K;iFAE5E,iC;MACqE,OAAA,SAAY,WAAU,UAAV,C;K;mFAEjF,2C;MACoF,OAAA,SAAY,WAAU,U  
AAV,EAA5B,QAA6B,C;K;4EAehG,0B;MAGuD,OAAA,SAAY,QAAO,GAAP,C;K;wEAEnE,4B;MAGgE,OAAA,  
SAAY,OAAM,KAAN,C;K;yFAK5E,2C;MACyF,OAAA,SAAY,SAAQ,OAAR,EAAiB,WAAjB,C;K;IAErG,iD;M  
AOKD,0B;QAAA,aAAsB,K;MACpE,IAAI,UAAJ,C;QACI,SAAS,SAAK,O;QACd,SAAS,KAAM,O;QACf,UTGG,  
MAAO,KSHM,ETGN,ESHU,ETGV,C;QSFV,IAAI,QAAO,CAAX,C;UAAc,OAAO,KAAK,EAAL,I;QACrB,iBA  
Ac,CAAd,UAA5B,GAAtB,U;UACI,eAAe,qBAAK,KAAL,C;UACf,gBAAGB,iBAAM,KAAN,C;UAEhB,IAAI,aA  
AY,SAAhB,C;YACI,WAAoB,cAAT,QAAS,C;YACpB,YAAsB,cAAV,SAAU,C;YAEtB,IAAI,aAAY,SAAhB,C;c  
ACwB,kBAAT,Q;cAAX,WDIO2C,gCAAY,cAfrB,YAAY,CAAZ,C;cCkPZ,kBAAV,S;cAAZ,YDnO2C,gCAAY,c  
AfrB,YAAY,CAAZ,C;cCoPIC,IAAI,aAAY,SAAhB,C;gBACI,OAAGB,iBAAT,QAAS,EAAU,SAAV,C;QAKhC  
,OAAO,KAAK,EAAL,I;QAEp,OAAO,4BAAU,KAAY,C;K;IAIf,4C;MAOqF,oCAAKB,KAAIB,C;K;IAErF,wD;  
MASI,OAAW,UAAJ,GACE,4BAAL,SAAK,EAA4B,KAA5B,CADF,GAGE,kBAAL,SAAK,EAAkB,KAAIB,C;K;  
IAIkD,oD;MAAU,OAAE,UAAF,CAAE,EAAU,CAAV,EAA0B,IAA1B,C;K;IAIvE,+C;MAAQ,oC;K;2F8G/SZ,oC  
;MACiF,O9G2Me,kB8G3ME,oBAAH,EAAG,C9G2MF,E8G3Mc,S9G2Md,C;K;mG8GzMHg,oC;MACqF,O9G2M  
e,sB8G3MM,oBAAH,EAAG,C9G2MN,E8G3MkB,S9G2MIB,C;K;I8GzMpG,mD;MAIoD,0B;QAAA,aAAsB,K;M  
ACtE,IAAI,CAAC,UAAL,C;QACI,O9GsMqF,qB8GtM7D,M9GsM6D,E8GtMrD,C9GsMqD,C;Q8GpMrF,OAAO,  
yBAAc,CAAd,EAAiB,MAAjB,EAAyB,CAAZB,EAA4B,MAAO,OAAnc,EAA2C,UAA3C,C;K;IAGf,iE;MAIqE,0  
B;QAAA,aAAsB,K;MACvF,IAAI,CAAC,UAAL,C;QACI,O9G2LqF,qB8G3L7D,M9G2L6D,E8G3LrD,U9G2LqD,  
C;Q8GzLrF,OAAO,yBAAc,UAAAd,EAA0B,MAA1B,EAAkC,CAAIC,EAAqC,MAAO,OAA5C,EAAoD,UAApD,  
C;K;IAGf,iD;MAIkD,0B;QAAA,aAAsB,K;MACpE,IAAI,CAAC,UAAL,C;QACI,O9GmLoE,mB8GnL9C,M9Gm  
L8C,C;Q8GjLpE,OAAO,yBAAc,mBAAS,MAAO,OAAb,IAAd,EAA5C,MAAtC,EAA8C,CAA9C,EAAiD,MAA  
O,OAAxD,EAAgE,UAAhE,C;K;IAGf,mC;MAGI,aACa,S9G0L2D,O8G1LhD,K9G0LgD,C;M8GzLxE,OAAO,kB  
AAkB,MAAO,OAAP,KAAe,C;K;IAG5C,4B;MAKOd,gCAAU,C;MAAV,U;QAAuB,kBAAR,yB;QAAQ,c;;UrH2n  
DvD,U;UADhB,IAAI,OCAAsB,qBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,+B;UAAhB,OAAGB,gBAAhB,C;Y  
AAgB,2B;YAAM,IAAI,CqH3nD4D,aAAT,qBrH2nDxC,OqH3nDwC,CAAS,CrH2nDhE,C;CAAYB,aAAO,K;cAAP  
,e;;UAC/C,aAAO,I;;QqH5nDgE,iB;;MAAvB,W;K;IAEpD,gD;MASiD,0B;QAAA,aAAsB,K;MAOxC,Q;MAN3B,  
IAAI,iBAAJ,C;QAAkB,OAAO,a;MACzB,IAAI,aAAJ,C;QAAMb,OAAO,K;MAC1B,IAAI,CAAC,UAAL,C;QAAi  
B,OAAO,kBAAQ,KAAR,C;MAExB,IAAI,SAAK,OAAL,KAAe,KAAM,OAazB,C;QAAiC,OAAO,K;MAEb,OA  
AL,SAAK,O;MAA3B,iBAAc,CAAd,wB;QACI,eAAe,qBAAK,KAAL,C;QACf,gBAAGB,iBAAM,KAAN,C;QACH  
B,IAAI,CAAU,SAAT,QAAS,EAAO,SAAP,EAAkB,UAAIB,CAAd,C;UACI,OAAO,K;MAIf,OAAO,I;K;IAIX,sF  
;MACkH,0B;QAAA,aAAsB,K;MACpI,oCAAKB,UAAIB,EAA8B,KAA9B,EAAqC,WAArC,EAAkD,MAAID,EA  
A0D,UAA1D,C;K;IAGJ,+B;MAYI,OxGmMmD,mBAAS,CwGnM5D,G9GwH4F,oB8GxHzD,C9GwHyD,E8GxHt  
D,C9GwHsD,CAvC9B,c8GjFrC,G9GqHoD,oB8GrHZ,C9GqHY,C8GrH7E,GAAyE,S;K;IAG7E,iC;MASI,OxGuL  
mD,mBAAS,CwGvL5D,G9G4G4F,oB8G5GzD,C9G4GyD,E8G5GtD,C9G4GsD,CAIB9B,c8G1FrC,G9GyGoD,oB  
8GzGZ,C9GyGY,C8GzG7E,GAAyE,S;K;IAG7E,8B;MAOiB,IAAN,I;M3H/FP,IAAI,E2H8FI,KAAK,C3H9FT,CA  
AJ,C;QACI,c2H6Fc,oD;Q3H5Fd,MAAM,gCAAYB,OAAQ,WAAjC,C;;M2H6FH,QAAM,CAAN,C;aACH,C;UAA  
K,S;UAAL,K;aACA,C;UAAU,OAAL,SAAK,W;UAAV,K;;UAEI,aAAa,E;UACb,IAAI,ExGgKoC,qBAAU,CwGh  
K9C,CAAJ,C;YACI,QAAQ,SAAK,W;YACb,YAAY,C;YACZ,OAAO,IAAP,C;cACI,IAAI,CAAC,QAAU,CAAX,  
MAAiB,CAArB,C;gBACI,UAAU,C;;cAEd,QAAQ,UAAW,C;cACnB,IAAI,UAA5B,CAAb,C;gBACI,K;;cAEJ,KAA

K,C;;;UAGb,OAAO,M;;MAnBf,W;K;IAwBJ,4D;MAOqE,0B;QAAA,aAAsB,K;MACvF,O9GkFiG,kB8GIFnF,W  
AAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAAgB,KAAhB,GAA2B,IAA1D,C9GkFmF,E8GIFB,6BA  
AM,iCAAwB,QAAxB,C9GkFY,C;K;I8GhFrG,4D;MAM+D,0B;QAAA,aAAsB,K;MACjF,O9GyEiG,kB8GzEnF,  
WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,UAAJ,GAAgB,KAAhB,GAA2B,IAApE,C9GyEm  
F,E8GzEA,oBAAR,OAAQ,C9GyEA,C;K;I8GvErG,iE;MAC0E,0B;QAAA,aAAsB,K;MAC5F,O9GqEiG,kB8GrEn  
F,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAAgB,IAAhB,GAA0B,GAAzD,C9GqEmF,E8GrEpB,  
6BAAM,iCAAwB,QAAxB,C9GqEc,C;K;I8GnErG,iE;MACoE,0B;QAAA,aAAsB,K;MACtF,O9GiEiG,kB8GjEnF,  
WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,UAAJ,GAAgB,IAAhB,GAA0B,GAAne,C9GiEm  
F,E8GjEF,oBAAR,OAAQ,C9GiEE,C;K;I+G7OrG,kD;MAEI,IAAI,gBAAJ,C;QAAsB,MAAM,6BAAyB,qCAAkC,  
QAAQ,CAAR,IAAIC,CAAZB,C;MAC5B,OAAO,CAAC,IAAD,I;K;IAGX,iF;MAQI,IAAI,EAAS,KAAT,oBAAiB,  
KAAjB,KAA2B,SAAS,QAAxC,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,UAA  
U,kBAAO,KAAP,C7GwBgC,I;M6GvB1C,IAAI,EAAQ,KAAR,kBAAGB,KAAhB,CAAJ,C;QACI,OAAO,UAAU,  
CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAO,SAAW,CAAC,OAAS,IAAV,KAAqB,EAhC,IAAwC,  
MAAQ,I;K;IAG3D,yE;MAQI,IAAI,SAAU,EA AV,MAAkB,CAAI B,IAAuB,SAAS,QAApC,C;QACI,OAAO,UAA  
U,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,GA AV,  
MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAQ,SAAU,CAAX  
,GAAkB,KAAIB,GAA4B,I;K;IAGvC,yE;MASI,IAAI,SAAS,QAAb,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,  
EAAoB,gBAApB,C;;MAGX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,EA AV,MAAiB,CAArB,C;QA  
CI,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER  
,IAAI,SAAU,EA AV,MAAiB,EAAR B,C;QACH,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CA  
AV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACH,OAAO,UAAU,CAA  
V,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,  
EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,  
GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,OAAQ,SAA  
U,EAAX,GAAoB,SAAU,CAA9B,GAAqC,KAArC,GAA+C,O;K;IAG1D,yE;MASI,IAAI,SAAS,QAAb,C;QACI,U  
AAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGJ,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,EA  
AV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GA AV,KAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,  
EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EA AV,MAAiB,CAArB,C;QACH,IAAI,SAAU,GA AV,MAAkB,GAAtB,  
C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EA AV,IAAgB,CAApB,C;Q  
ACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;aACJ,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QA  
CH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI  
,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C  
;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;  
;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;M  
AEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OA  
AO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAQ,SAAU,EAAX,GAAoB,SAAU,EA A9B,GA  
AuC,SAAU,CAAjD,GAAwD,KAAxD,GAAkE,O;K;;;IAmB7E,oE;MAkB0B,UAGJ,MAHI,EAKJ,MALI,EAMJ,M  
ANI,EASJ,MATI,EAUJ,MAVI,EA WJ,MAXI,EA gBA,MAhBA,EAiBA,MAjBA,EAKBA,MAiBA,EAoBA,MApBA  
,EAqBA,OArBA,EASBA,OA tBA,EAuBA,O;M5H9JtB,IAAI,E4HgII,cAAc,CAAd,IAAmB,YAAY,MAAO,OAAtC  
,IAAgD,cAAc,Q5HhII,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;M4HgIV,YAAY,cA  
AU,CAAC,WAAW,UAA X,IAAD,IAA0B,CAA1B,IAAV,C;MACZ,gBAAgB,C;MACHb,gBAAgB,U;MAEhB,OA  
AO,YAAY,QAA nB,C;QACI,WAAW,mBAAO,gBAAP,EAAO,wBAAP,Q7G1H2B,I;Q6G4HIC,WAAO,GAAP,C;  
UACI,MAAM,kBAAN,EAAM,0BAAN,YAA0B,OAAL,IAAK,C;eAC9B,WAAO,IAAP,C;UACI,MAAM,kBAAN,  
EAAM,0BAAN,YAA4C,OAAR B,QAAS,CAAV,GAAgB,GAAM,C;UAC5C,MAAM,kBAAN,EAAM,0BAAN,YA  
A+C,OAAXB,OAAS,EA AV,GAAM B,GAAM,C;eAEnD,WAAO,KAAP,IAAiB,QAAQ,KAAzB,C;UACI,MAAM,k  
BAAN,EAAM,0BAAN,YAA6C,OAAtB,QAAS,EA AV,GA AiB,GAAM,C;UAC7C,MAAM,kBAAN,EAAM,0BA  
AN,YAAuD,OA A/B,QAAS,CAAV,GA AiB,EA AiB,GAA2B,GAAM,C;UACvD,MAAM,kBAAN,EAAM,0BAAN,  
YAA+C,OAAXB,OAAS,EA AV,GAAM B,GAAM,C;;UAG/C,gBAAgB,uBAAuB,MAAvB,EAA+B,IAA/B,EAAqC,

SAArC,EAAGD,QAaHd,EAA0D,gBAA1D,C;UACHB,IAAI,aAAa,CAAjB,C;YACI,MAAM,kBAAN,EAAM,0BAA  
AN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAAN,YAAqB,0BAA0B,CAA1B,C;YACrB,  
MAAM,kBAAN,EAAM,0BAAAN,YAAqB,0BAA0B,CAA1B,C;;YAErB,MAAM,kBAAN,EAAM,0BAAAN,YAAkD  
,OAA3B,aAAc,EAaf,GAAsB,GAAM,C;YACID,MAAM,mBAAN,EAAM,2BAAN,aAA6D,OAArC,aAAc,EAaf,  
GAAuB,EAAXB,GAAiC,GAAM,C;YAC7D,MAAM,mBAAN,EAAM,2BAAN,aAA4D,OAApC,aAAc,CAaf,GAA  
sB,EAavB,GAAgC,GAAM,C;YAC5D,MAAM,mBAAN,EAAM,2BAAN,aAAoD,OAA7B,YAAc,EAaf,GAawB,  
GAAM,C;YACpD,6B;;;MAMhB,OAAW,KAAM,OAAN,KAAc,SAA1B,GAA6B,KAA7B,GAA8C,UAAAN,KAA  
M,EAAO,SAAP,C;K;;IAQzD,mE;MAiByB,Q;M5H9LrB,IAAI,E4HwLI,cAAc,CAAd,IAAmB,YAAY,KAAM,OA  
ArC,IAA6C,cAAc,Q5HxL/D,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;M4HwLV,gBA  
AgB,U;MACHB,oBAAoB,sB;MAEpB,OAAO,YAAY,QAAnB,C;QACI,WAAW,KAAMB,CAAb,gBAAa,EAAb,w  
BAAa,O;QAE1B,YAAQ,CAAR,C;UACI,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;aACIB,YAAS,CAAT,KAAc,EA  
Ad,C;UACI,WAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IA  
AI,QAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,IAAD,IAAb,K;;YAEA,aAAc,gBAAY,  
OAAL,IAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,  
EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,g  
BAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,MAAK,CAAZ,C;YACd,w  
BAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,  
EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAA  
a,CAAC,MAAD,IAAb,K;;YAEA,WAAW,MAAD,GAAQ,KAAR,IAAQB,EAARB,GAA2B,K;YACtC,UAAW,SAA  
S,IAAV,GAAoB,K;YAC9B,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,aAAc,gBAAW,OAAJ,GAAI,CAAX,C;  
YACd,wBAAa,CAAb,I;;UAIJ,UAAU,CAAV,EAAa,SAAb,EAAwB,gBAAxB,C;UACA,aAAc,gBAAO,gBAAP,C  
;;MAK1B,OAAO,aAAc,W;K;ICtQzB,uC;MAU2D,OAAwB,CAAXB,2BAAwB,mBAAS,SAAT,C;K;IAEnF,oC;M  
AKI,OAAQ,OAAW,mBAAL,SAAK,CAAX,C;K;IAGZ,6C;MAMI,IAAI,cAAS,SAAb,C;QACI,iBAAsB,SAAY,Y;  
QACIC,IAAI,kBAAJ,C;UACS,SAAL,eAA+B,iBAAc,SAAd,E;;UAE/B,UAAW,WAAL,SAAJ,C;;K;IAUnB,6C;M  
AC4B,UAAjB,M;MAAP,OAAO,WAAiB,OAAsB,SAAY,YAAjB,4CAA+D,W;K;IAI9E,iC;MACI,gBAAqB,sB;MA  
CrB,iBAAsB,E;MACTB,kBAA+B,E;MAC/B,uBAAiC,C;K;uDAEjC,qB;MACc,qBAAV,SAAU,EAAC,EAAd,EAA  
kB,EAAIB,C;MACV,OAAO,aAAO,W;K;gDAG1B,qB;MAA6D,gBAAR,c;MAAQ,c;;Q3I41Y7C,Q;QAaHb,wBA  
AgB,SAaHb,gB;UAAgB,cAAA,SAaHb,M;UAAAsB,IAAc,O2I51Y+B,c3I41Y7C,C;YAAwB,aAAO,I;YAAP,e;;Q  
AC9C,aAAO,K;;M2I71Y8C,iB;K;sDAErD,wC;MACI,KAAK,qBAAL,SAAK,EAAC,MAAd,EAAsB,SAATB,CAA  
L,C;QAAyC,M;MAEzC,YAAY,SAAK,M;MACjB,OAAO,aAAP,C;QACI,KAAM,qBAAN,KAAM,EAAC,MAAd,  
EAAsB,aAATB,CAAN,C;UAA8C,M;QAC9C,QAAQ,KAAM,M;;K;sDAITB,wC;MASgB,IAAiB,IAAjB,EA2BE,M;  
MAnCd,aAAO,gBAAO,MAAP,CAAE,gBAAO,SAAP,C;MACTB,gBAAgB,SAAK,W;MACrB,IAAI,eAAQ,SAAR,  
CAAJ,C;QACI,aAAO,gBAAO,kCAAP,CAA2C,gBAAO,SAAP,CAAKB,gBAAO,KAAP,C;QACpE,OAAO,K;;MA  
EH,cAAY,MAAK,SAAL,C;MAEpB,YAAY,CAAiB,OAAsB,SAAY,MAAjB,2D;MACZ,IAAI,aAAJ,C;Q1HyBG,S0  
HxBwB,WAAN,KAAM,EAQ,SAAR,C;QAAvB,iBAAoD,KAAK,CAAT,GAAY,CAAZ,GAAMB,KAAe,gBAaf,  
I;QACnE,IAAI,eAAc,CAAIB,C;UAAqB,aAAO,gBAAO,SAAP,CAAKB,gBAAO,IAAP,C;QAC9C,IAAI,e1G8Mo  
C,YAAU,C0G9MID,C;UACI,kBAAW,K;UACX,uBAAgB,U;;UAEhB,QAAQ,wBAAiB,KAAjB,EAAwB,UAAxB,  
C;;QAEZ,IAAI,M1GgNuC,UAAAS,C0GhNpD,C;UAEuB,U;UAAA,IAAI,eAAc,CAAIB,C;YAAA,SAAqB,C;;Y3Gq  
+BpC,U;YADhB,YAAY,C;YACI,oB2Gr+B+C,S3Gq+B/C,C;YAAhB,OAAgB,gBAAhB,C;cAAgB,sC;cAAM,I2Gr  
+BgE,U3Gq+BID,oB2Gr+BkD,MAAK,E3Gq+BrE,C;gBAAwB,qB;;Y2Gr+Bf,SAA4B,I3Gs+BpD,K2Gt+BoD,I;;U  
AA/C,yB;U7GorCC,kB;UADb,YAAY,C;UACC,S6GnrCK,aAAN,KAAM,C7GmrCL,W;UAAb,OAAa,gBAAb,C;  
YAAa,wB;Y6GlrCG,I7GkrCU,oBAAmB,cAAnB,EAAMB,sBAAnB,U6GlrCN,gBAAJ,C;cAA2B,aAAO,uB;YACI  
C,aAAO,gB7GirCgC,I6GjrChC,CAAa,gBAAO,IAAP,C;;UAGxB,aAAO,gBAAO,KAAP,CAAc,gBAAO,IAAP,C;;  
;QAGzB,aAAO,gBAAO,SAAP,CAAKB,gBAAO,IAAP,C;;MAG7B,iBAAiB,mC;MACjB,IrIuHoD,CqIvHhD,UrIu  
HiD,UqIvHrD,C;QACI,uBAAuB,SAAS,M;QACTB,8B;QAAV,OAAU,gBAAV,C;UAAU,qB;UACJ,qBAAF,CAAE  
,EAAC,gBAAd,EAAgC,cAAhC,C;;MAGV,OAAO,I;K;yDAGX,6B;MAIwB,Q;MAHpB,mBAAwB,C;MACxB,gB  
AAqB,C;MACrB,mBAAwB,C;MACJ,OzHyIjB,MAAO,KyHzIgB,eAAS,OAAT,GAAKB,oBAAIB,IzHyIhB,EyHzI  
iD,KAAM,OAAN,GAae,UAAf,IzHyIjD,C;MyHzIV,eAAY,CAAZ,oB;QACI,QAAQ,iBAAY,iBAAN,KAAM,CA

AN,GAaKB,GAAIB,IAAN,C;QACR,IAAI,MAAK,2BAaKB,iBAAT,eAAS,CAAT,GAAqB,GAARb,IAAT,CAAT,  
C;UAA6C,K;QAC7C,IAAI,MAAK,EAAT,C;UACI,8BAAGb,CAAhB,I;UACA,eAAe,S;UACf,YAAAY,G;;;MAGp  
B,IAAI,gBAAGb,CAApB,C;QAAuB,OAAO,K;MAC9B,OAAO,eAAe,CAAf,IAAoB,iBAAY,iBAAN,KAAM,CA  
AN,IAAmB,YAAAnB,GAaKc,CAaIC,KAAN,MAA+C,EAA1E,C;QACI,8BAAGb,CAAhB,I;MAGJ,OAAa,YAAN,  
KAAM,EAAS,YAAT,CAAN,IAA+B,cAAW,eAAe,CAAf,IAAX,uCAA/B,C;K;;yHC/H+C,Y;MAAQ,W;K;IAEtE,  
gD;MACkB,UAMP,M;MANO,IAAI,aAAY,CAAhB,C;QACV,Y;;QAEA,UxBsY8C,MAAW,KwBtY/C,IxBsY+C,  
EwBtYtC,QxBsYsC,C;QwBrYzD,OAAA,IAAO,OxB2UmC,MAAW,KwB3UpC,KxB2UoC,CwB3UxC,GAAa,GA  
AnB,CAAP,GAAiC,GAAjC,GxBwV2C,MAAW,MwBxVV,KxBwVU,C;;MwB5V1D,kB;MAMO,IxBYUuC,MAA  
W,KwBzU1C,OxBYU0C,CwBzU9C,GAAe,MAAnB,C;QAEmC,SAA9B,OAAAY,SAAQ,QAAR,C;;QAGpB,exBoU  
0C,MAAW,KwBpUIC,OxBoukC,C;QwBnUrD,qBAA8B,QAAAY,axBgRC,MAAW,MAvCV,MAAW,OwBzOU,Qx  
ByOV,CAuCD,CwBhRA,GAAwB,QAAPC,C;QAC1C,SAAI,UAAU,CAAd,GAAiB,MAAG,cAAPB,GAAyC,c;;M  
AP7C,a;K;IAWJ,6C;MACI,OAAa,KAAY,gBAaE,OAAf,EAawB,MAAK,4BAA2B,QAA3B,CAAL,EAaxB,C;K;I  
CtBQ,4C;MAFrC,e;MAEsC,0B;MAFtC,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAGI,uEAGY,GAHZ,C;MAIA  
,yEAGa,MAHb,C;MAIA,yEAGa,SAHb,C;MAIA,+DAGQ,KAHR,C;MAIA,+DAGQ,MAHR,C;MAIA,2DAGM,M  
AHN,C;MAIA,yDAGK,OAHL,C;K;;IAxBA,gD;MAAA,yB;MAAA,wC;K;;IAIA,iD;MAAA,yB;MAAA,yC;K;;IAI  
A,iD;MAAA,yB;MAAA,yC;K;;IAIA,4C;MAAA,yB;MAAA,oC;K;;IAIA,4C;MAAA,yB;MAAA,oC;K;;IAIA,0C;  
MAAA,yB;MAAA,kC;K;;IAIA,yC;MAAA,yB;MAAA,iC;K;;IA3BJ,+B;MAAA,4Q;K;;IAAA,oC;MAAA,a;AAA,  
a;UAAA,6C;aAAA,c;UAAA,8C;aAAA,c;UAAA,8C;aAAA,S;UAAA,yC;aAAA,S;UAAA,yC;aAAA,O;UAAA,uC;  
aAAA,M;UAAA,sC;;UAAA,6D;;K;;IAiCA,4D;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAA  
W,cAARb,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMb,UAAW,cAAvC,C;WAC3B,0B  
AAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMb,UAAW,cAAvC,C;;QACnB,Y;MAHZ,W;K;IAOJ,oE;M  
AGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAARb,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,  
sBAA8C,uBAARc,UAAW,cAAX,GAAMb,UAAW,cAAO,CAA9C,C;WAC3B,0BAAsB,CAAtB,C;QAA2B,iBAA  
8C,uBAARc,UAAW,cAAX,GAAMb,UAAW,cAAO,CAA9C,C;;QACnB,Y;MAHZ,W;K;IAOJ,8D;MAGW,Q;MA  
DP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAARb,C;MAEvC,0BAAsB,CAAtB,C;QACI,YAAkD,uBAA  
rC,UAAW,cAAX,GAAMb,UAAW,cAAO,C;QACID,aAAa,eAAQ,KAAR,C;QAET,sBAAS,KAAT,GAaKB,KAAL  
B,E;UAA2B,a;AC3B,uBAAQ,CAAR,C;;;aAIR,0BAAsB,CAAtB,C;QAA2B,iBAA8C,uBAARc,UAAW,cAAX,G  
AAmB,UAAW,cAAO,CAA9C,C;;QACnB,Y;MAXZ,W;K;ICrDJ,+B;MAAA,mC;MAUuB,wB;MALf,aAAR,OAA  
O,OAAQ,KAAL,WAAY,IAAG,OAAO,SAAX,IAAwB,CAAC,CAAC,OAAO,SAAS,K;MADpE,sBAGQ,MAHR,G  
AIQ,iBAAa,OAAb,CAJR,GAMQ,qBAAW,OAAx,IAAA,4GACO,+B;K;4CAIf,Y;MAAmC,OAAA,mBAAa,U;K;;  
;IAfpD,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAwB2B,+B;MAAC,sB;K;IAEW,+D;MAAA,0C;MAAS,mB;M  
ACxC,iBAAgB,yBAAQ,S;K;8DACxB,Y;M7HyEG,Q6HxEC,8BAAQ,QAAO,cAAP,C;MAAyB,c9IZIC,EAAL,CA  
AJ,C;M8IY2C,Y9IuF3C,EAAL,CAAJ,C;M8IvFC,OAA4D,aAAR,OAAQ,qCAAR,aAAiD,aAN,KAAM,yCAAjD,  
C;K;;qCAH5D,Y;MAAmC,mD;K;sCAMnC,Y;MAAkC,qC;K;;IAKF,4C;MAAiC,4E;MAAhC,8B;K;2CACjC,Y;M  
AA8B,OAAA,gBAAY,M;K;;CAC1C,Y;MAAkC,2C;K;;IAGtC,6B;MAAA,iC;MAEoC,4E;K;uCACHc,Y;MAA8B  
,OAAe,U;K;2CAC7C,Y;MAAkC,+B;K;;IAJtC,yC;MAAA,wC;QAAA,uB;;MAAA,iC;K;IC1CA,gD;MAQ+B,kBA  
ApB,wBAAc,IAAd,C;MAA0B,I9HgEjC,a;M8HhEA,O9HiEO,W;K;I8H9DX,gD;MAQqD,kBAA1B,gBAAhB,sCA  
AgB,EAAC,IAAd,EAAoB,IAApB,C;MAAiC,sB9HoEID,W8HpEkD,C;MAAxD,O9HqEO,W;K;I+HzFX,yC;MAEK  
D,8B;MAAA,OCGN,aDhwB,yBAAa,QAAb,mCCGxB,ChH+xBgC,sB;K;I+GhyB5E,2C;MhJggIW,kBAAY,gB;M  
AoGH,Q;MAAhB,wBgJ7IIqB,UhJ6IIrB,gB;QAAgB,cgJ7IIK,UhJ6IIrB,M;QAAsB,IAAI,CgJ7IIkB,sBhJ6IIP,OgJ7II  
O,ChJ6IIrB,C;UAAyB,WAAY,WAAL,OAAJ,C;;MgJ7II3D,qBhJ8IIO,W;MgJ7IIP,I1IgNwD,C0IhNpD,c1IgNqD,U0  
IhNzD,C;Q/GgKuC,U;Q+G/JnC,qB/G+JyD,OAAtB,+B+G/Jd,mB/G+Jc,uBAAsB,CAAO,W;QuGkO7C,kBAAhB,s  
B;QQ/XC,0C;QACA,IAAI,E/G8QoC,0BAAU,C+G9Q9C,CAAJ,C;UACI,2BAAO,GAAP,C;;QAEW,sCAAa,GAA  
b,C;QALnB,sB/H4DG,WuHoUqC,W;QQzXxC,OAAO,I;;MAGX,OAAO,K;K;IAGX,8C;MAOmB,c;;QhJi3YC,Q;  
QAAhB,wBgJj3YI,UhJi3YJ,gB;UAAgB,cgJj3YZ,UhJi3YJ,M;UAAsB,IgJj3YD,sBhJi3Ye,OgJj3Yf,ChJi3YC,C;YA  
AwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MgJl3YP,e;QACI,kBAA6B,MAAX,UAAW,C;Q/GyIM,U;Q+GxIb,a/  
GwImC,OAAtB,+B+GxIvB,mB/GwIuB,uBAAsB,CAAO,W;Q+GxIX,kBC/BjB,aD+BD,MC/BC,ChHg1C6C,uBA  
AzB,CAAYB,C;QbnmB9E,kBAAS,gB;QA2FA,U;QAAA,+B;QAAhB,OAAgB,gBAAhB,C;UAAgB,6B;UAAM,I4



HzyB4C,4B5HyyB9B,S4HzyB8B,C5HyyB5C,C;YAAwB,WAAY,WAAl,SAAJ,C;:Q4HzyBtD,sBAAmF,e5H0yBh  
F,W4H1yBgF,EAAa,GAAb,C;QACnF,OAAO,I;:MAGX,OAAO,K;K;IEnCP,iC;MAAQ,8BAAY,IAAK,UAAjB,IA  
A8B,uBAAY,IAAK,mB;K;IAOvD,oC;MAAQ,8BAAY,IAAK,a;K;ICZ7B,4B;MAGI,OAAO,yBAAP,C;QACI,sBA  
AY,mCAAZ,C;:K;IAIR,uC;MAOI,sBAAY,sCAAgB,gBAAE,IAAf,CAA5B,C;MACA,OAAO,S;K;ICbP,4B;MAA  
Q,mB;K;IACR,mC;MACI,eAAO,K;K;IAKX,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;iHCoBf,sJ;MAEyC,q  
B;QAAA,QAaKb,I;MAAM,qB;QAAA,QAaKb,I;MAAM,uB;QAAA,UAAoB,K;MAAO,yB;QAAA,YAAsB,I;MA  
AM,kC;QAAA,qBAA+B,I;MAAM,qC;QAAA,wBAaKc,K;MAAO,+C;QAAA,kCAA4C,K;MAAO,4C;QAAA,+B  
AAyC,K;MACtT,QAAQ,E;MACR,EAAE,OAAf,IAAa,K;MACb,EAAE,OAAf,IAAa,K;MACb,EAAE,SAAF,IAA  
e,O;MACf,EAAE,WAAf,IAAiB,S;MACjB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,uBAAF,IAA6B,qB;MAC7  
B,EAAE,iCAAF,IAAuC,+B;MACvC,EAAE,8BAAF,IAAoC,4B;MACpC,OAAO,C;K;+Gaw0BX,wD;MAEwC,6B  
;QAAA,gBAAYB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;M  
AC/I,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAKB,U;MA  
CIB,EAAE,UAAf,IAAgB,Q;MACHB,OAAO,C;K;6EA6CX,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7D,mC;MAEo  
E,UAAy,KAAZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEeQ,UAAy,KAAZ,IA  
AqB,K;K;6EAuB1F,4B;MAEeQ,iBAAY,KAAZ,C;K;6EAErE,mC;MAE4E,UAAy,KAAZ,IAAqB,K;K;6EAuBjG,  
4B;MAE+D,iBAAY,KAAZ,C;K;6EAE/D,mC;MAEsE,UAAy,KAAZ,IAAqB,K;K;6EAuB3F,4B;MAEgE,iBAAY,  
KAAZ,C;K;6EAEhE,mC;MAEuE,UAAy,KAAZ,IAAqB,K;K;6EAuB5F,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7  
D,mC;MAEoE,UAAy,KAAZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEeQ,UAA  
y,KAAZ,IAAqB,K;K;6EAuB1F,4B;MAEiE,iBAAY,KAAZ,C;K;6EAEjE,mC;MAEwE,UAAy,KAAZ,IAAqB,K;  
K;6EAuB7F,4B;MAEeK,iBAAY,KAAZ,C;K;6EAEIE,mC;MAEyE,UAAy,KAAZ,IAAqB,K;K;6GC3oC9F,wD;M  
AEqC,6B;QAAA,gBAA+B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WA  
AqB,K;MACpJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAA  
kB,U;MACIB,EAAE,UAAf,IAAgB,Q;MACHB,OAAO,C;K;mIAiCX,+B;MAEgD,mC;QAAA,sBAAGC,K;MAC5  
E,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,OAAO,C;K;4EC9CX,4B;MAEgE,iBAAY,KAAZ,C;K;4E  
AgChE,4B;MAEyE,iBAAY,KAAZ,C;K;4EAIbZE,4B;MAEmE,iBAAY,KAAZ,C;K;4EAYnE,4B;MAE0E,iBAA  
Y,KAAZ,C;K;oIC7a1E,4H;MAE8C,qB;QAAA,QAaiB,E;MAAI,6B;QAAA,gBAAGC,E;MAAW,iC;QAAA,oBAA  
2D,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,qC;QAAA,wBAmJvJ,U;:MANJqO,+B;QAAA,kBAmJrO,U;:MANJ6  
S,4B;QAAA,eAA+B,S;MAC3a,QAAQ,E;MACR,EAAE,OAAf,IAAa,K;MACb,EAAE,eAAF,IAAqB,a;MACrB,E  
AAE,mBAAF,IAAyB,iB;MACzB,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,i  
BAAF,IAAuB,e;MACvB,EAAE,cAAF,IAAoB,Y;MACpB,OAAO,C;K;wIAYX,mC;MAEgD,2B;QAAA,cAAuB,E;  
MAAI,0B;QAAA,aAAsB,E;MAC7F,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,YAAF,IAAKB,U;  
MACIB,OAAO,C;K;8HAKEX,+D;MAEgG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAA  
A,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,SA  
AF,IAAe,O;MACf,EAAE,YAAF,IAAKB,U;MACIB,EAAE,UAAf,IAAgB,Q;MACHB,OAAO,C;K;4HAwBX,iE;M  
AE0C,4B;QAAA,eAAwB,E;MAAI,wB;QAAA,WAAyB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAA  
uB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAf,IA  
AgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAKB,U;MACIB,EAAE,UAAf,IAAgB,Q;MACHB,  
OAAO,C;K;sGAUqE,qB;MAAQ,OAAW,U;K;sGAEnB,qB;MAAQ,OAAW,U;K;4GAehB,qB;MAAQ,OAAc,a;K;  
wGAS1B,qB;MAAQ,OAAy,W;K;0HAEX,qB;MAAQ,OAAqB,oB;K;kGASnD,qB;MAAQ,OAAQ,Q;K;oGAehB,q  
B;MAAQ,OAAU,S;K;sGAEjB,qB;MAAQ,OAAW,U;K;wHAEV,qB;MAAQ,OAAoB,mB;K;wHAE5B,qB;MAAQ,  
OAAoB,mB;K;kHAE/B,qB;MAAQ,OAAiB,gB;K;kHAEzB,qB;MAAQ,OAAiB,gB;K;oHASd,qB;MAAQ,OAAkB,  
iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;wIAEHb,qB;MAAQ,OAA4B,2B;K;4  
FC1MnI,uD;MAE8B,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAaE,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;  
QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,  
QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAKB,U;MACIB,EAAE,UAAf,IAAgB,Q;M  
ACHB,OAAO,C;K;kGAuBX,sE;MAEiC,6B;QAAA,gBAA8B,I;MAAM,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,S  
AAE,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvL,QAA  
Q,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,S

AAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGA8DX,8U;MAEiC,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC3wB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wGAgDX,kQ;MAEoC,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7IB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGAsCX,iX;MAEiC,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,yB;QAAA,YAAkB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACr2B,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,WAAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGA2BX,0E;MAEiC,oB;QAAA,OAAgB,E;MAAI,2B;QAAA,cAAwB,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtM,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,MAAF,IAAY,I;MACZ,EAA

E,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wGAmDX,4S;MAEoC,mB;QAAA,MAAe,E;MAAI,oB;QAAA,OAAgB,E;MAAI,wB;QAAA,WAAiB,C;MAAG,sB;QAAA,SAAmB,K;MAAO,2B;QAAA,cAAwB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjtB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAC hB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;8GAuBX,6D;MAEuC,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7K,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wECnbX,4B;MAEyE,iBAAAY,KAAZ,C;K;wEAEzE,2B;MAEgG,iBAAY,IAAZ,C;K;wEAwBhG,oC;MAE+F,UAAAY,KAAZ,IAAqB,M;K;wEAmFpH,2B;MAEqE,iBAAY,IAAZ,C;K;wEAErE,kC;MAE2E,UAAAY,IAAZ,IAAoB,K;K;wEAssC/F,4B;MAEyE,iBAAY,KAAZ,C;K;wEA0BzE,4B;MAEyE,iBAAY,KAAZ,C;K;wEAsBzE,4B;MAEuE,iBAAY,KAAZ,C;K;wEAyBvE,4B;MAE6E,iBAAY,KAAZ,C;K;2FA4C7E,gD;MAEiC,qB;QAAA,QAAiD,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACIK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uEA+UX,4B;MAEuE,iBAAY,KAAZ,C;K;wEAEvE,2B;MAE6F,iBAAY,IAAZ,C;K;wEAqN7F,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,oC;MAE2F,UAAAY,KAAZ,IAAqB,M;K;+FAuehH,wD;MAEmC,6B;QAAA,gBAA8B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uGAuIX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+HAyCX,iB;MAEmD,qB;QAAA,QAAkB,I;MACjE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+FA0MX,sE;MAEmC,oB;QAAA,OAAgB,I;MAAM,wB;QAAA,WA0+G4B,S;MA1+GwB,kB;QAAA,KAAc,E;MAAI,wB;QAAA,WAAoB,I;MAAM,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAoB,I;MAAM,qB;QAAA,QAAiB,I;MAAM,oB;QAAA,OAAgB,I;MACnP,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,IAAF,IAAU,E;MACV,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qIAGDX,iB;MAEsD,qB;QAAA,QAAkB,I;MACpE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+GAKBX,qB;MAE2C,yB;QAAA,YAAmB,S;MAC1D,QAAQ,E;MACR,EAAE,SAAF,IAAe,S;MACf,OAAO,C;K;wEAkCX,4B;MAEqF,iBAAY,KAAZ,C;K;yFAGCrF,4V;MAEgC,4B;QAAA,eAA8B,I;MAAM,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC9yB,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,S

AAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wEAwEX,2B;MAE+D,iBAAY,IAAZ,C;K;iGA2D/D,gD;MAEoC,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACII,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;qGA2BX,yD;MAEsC,sB;QAAA,SAAkB,E;MAAI,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC5J,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6GAuBX,oD;MAEOC,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;2FAoFX,kF;MAEiC,uB;QAAA,UAAmB,E;MAAI,wB;QAAA,WAAoB,E;MAAI,sB;QAAA,SAAe,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjN,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EA AE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;iHAyBX ,0D;MAEqE,sB;QAAA,SAAe,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACzK,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wEA mXX,4B;MAE kE,iBAAY,KAAZ,C;K;wEAEIE,2B;MAEOE,iBAAY,IAAZ,C;K;wEAUpE,4B;MAEsE,iBAAY,KAAZ,C;K;wEAE tE,2B;MAEW,E,iBAAY,IAAZ,C;K;wEAaxE,4B;MAE+D,iBAAY,KAAZ,C;K;wEAE/D,2B;MAEiE,iBAAY,IAAZ,C;K;mGA0CjE,8G;MAEqC,gC;QAAA,mBAooF8C,M;;MApoFe,gC;QAAA,mBAmpFT,S;;MANpFyE,oC;QAAA,uBA8pFjE,S;;MA9pF6I,2B;QAAA,cAAoB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,6B;QAAA,gBAyqFIO,K;;MA xqFvE,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,sBAAF,IAA4B,oB;MAC5B,EAAE,aAAF,IAAmB,W;MACnB,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,eAAF,IAAqB,a;MA CrB,OAAO,C;K;+FAwCX,mF;MAEmC,oB;QAAA,OAAa,I;MAAM,sB;QAAA,SAAkB,E;MAAI,2B;QAAA,cAA uB,E;MAAI,sB;QAAA,SAAY,C,I;MAAM,qB;QAAA,QAA6B,E;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA ,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACxQ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF ,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EA AAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6FA4 BX,2B;MAEkC,+B;QAAA,kBAA4B,K;MAC1D,QAAQ,E;MACR,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;2 FA2DX,iE;MAEiC,wB;QAAA,WAAqB,K;MAAO,oB;QAAA,OAAe,C;MAAG,sB;QAAA,SAAkB,E;MAAI,uB;Q AAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,U AAF,IAAgB,Q;MACHb,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MAC f,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;yFA8FX,6B;MAEgC,oB;QAAA, OA+7E6C,S;;MA/7EL,2B;QAAA,cCl2He,M;;MDm2HnF,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE, aAAF,IAAmB,W;MACnB,OAAO,C;K;wEAoDX,0B;MAE+D,iBAAY,GAZ,C;K;wEAE/D,iC;MAEqE,UAA Y,GA AZ,IAAmB,K;K;+FAoDxF,oF;MAEmC,mB;QAAA,MAAe,I;MAAM,wB;QAAA,WAAoB,I;MAAM,wB;QAAA ,WAAoB,I;MAAM,mB;QAAA,MAAe,E;MAAI,2B;QAAA,cAAwB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;Q AAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvO,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE, UAAF,IAAgB,Q;MACHb,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,KAAF,IAAW,G;MACX,EAAE,aAAF,IAAmB ,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAA O,C;K;iFAwNX,yC;MAE4B,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K; MACtG,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;

MACHB,OAAO,C;K;6FAwBX,iD;MAEkC,sB;QAAA,SA Ae,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IA Ae,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;uGASX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IA Ae,O;MACf,OAAO,C;K;6GAYX,kC;MAE0C,uB;QAAA,UAAoB,K;MAAO,oB;QAAA,OAAiB,K;MAAO,uB;QAAA,UAAoB,K;MAC7G,QAAQ,E;MACR,EAAE,SAAF,IA Ae,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IA Ae,O;MACf,OAAO,C;K;wEAkEX,4B;MAE6D,iBAAY,KAAZ,C;K;wEAU7D,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;uGAsCxE,oH;MAEuC,yB;QAAA,YAAsB,K;MAAO,0B;QAAA,aAAuB,S;MAAW,6B;QAAA,gBAA0B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,iC;QAAA,oBAA8B,S;MAAW,qC;QAAA,wBAaKc,S;MAAW,+B;QAAA,kBAaKc,S;MAC1R,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IA Ae,O;MACf,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;mGAgFX,oB;MAEqC,wB;QAAA,WAAqB,K;MACtD,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wEA+MX,2B;MAEiE,iBAAY,IAAZ,C;K;2GakCjE,c;MAEyC,kB;QAAA,KAAgB,S;MACrD,QAAQ,E;MACR,EAAE,IAAF,IAAU,E;MACV,OAAO,C;K;2FAuMX,gB;MAGl,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;wEAgBX,4B;MAEiE,iBAAY,KAAZ,C;K;wEAejE,oC;MAE4E,iBAAY,aAAZ,C;K;wEAuT5E,4B;MAEmE,iBAAY,KAAZ,C;K;uFA2CnE,sB;MAE+B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAC9F,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,OAAO,C;K;qFA0CX,+B;MAE8B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAakB,G;MACtG,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,OAAO,C;K;wEAOX,4B;MAEmE,iBAAY,KAAZ,C;K;yFAiHnE,oB;MAEgC,wB;QAAA,WAy2B+C,M;MAx2B3E,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6FAeX,+B;MAEkC,oB;QAAA,OAAgB,S;MAAW,mB;QAAA,MAAe,S;MAAW,wB;QAAA,Waq1BR,M;MAp1B3E,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GAwCX,yD;MAE0C,qB;QAAA,QAAiB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IA Ae,O;MACf,EAAE,SAAF,IA Ae,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;yGaiCX,mC;MAEwC,qB;QAAA,QA2wByD,Q;MA3wBK,sB;QAAA,SA2wBL,Q;MA3wBoE,wB;QAAA,WA4vBtF,M;MA3vB3E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;2FAYX,2B;MAEiC,mB;QAAA,MAuwB0C,Q;MAvwBJ,0B;QAAA,aAAsB,S;MACzF,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,YAAF,IAAkB,U;MACIB,OAAO,C;K;+GAYX,0B;MAE2C,uB;QAAA,UaqvBgC,Q;MARvBU,qB;QAAA,QAqvBV,Q;MApvBvE,QAAQ,E;MACR,EAAE,SAAF,IA Ae,O;MACf,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;wEAgCX,4B;MAE+D,iBAAY,KAAZ,C;K;qFAyaY,qB;MAAQ,OAAU,S;K;6FAEd,qB;MAAQ,OAAc,a;K;uFAEzB,qB;MAAQ,OAAW,U;K;iFASxB,qB;MAAQ,OAAE,G;K;iFAEX,qB;MAAQ,OAAQ,O;K;uFAEb,qB;MAAQ,OAAW,U;K;uFAS3B,qB;MAAQ,OAAW,U;K;mFAErB,qB;MAAQ,OAAQ,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;yFAShB,qB;MAAQ,OAAW,W;K;uFAErB,qB;MAAQ,OAAW,U;K;+FAEf,qB;MAAQ,OAAe,c;K;uFAE3B,qB;MAAQ,OAAW,U;K;uFAEnB,qB;MAAQ,OAAW,U;K;mFASrB,qB;MAAQ,OAAQ,Q;K;iFAEiB,qB;MAAQ,OAAQ,O;K;6EAEiB,qB;MAAQ,OAAW,U;K;uFAET,qB;MAAQ,OAAW,U;K;qFASiB,qB;MAAQ,OAAU,S;K;qFAEiB,qB;MAAQ,OAAU,S;K;6EASr,qB;MAAQ,OAAW,U;K;mFAEX,qB;MAAQ,OAAQ,Q;K;+EAEEnB,qB;MAAQ,OAAO,M;K;+EAS/B,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAQ,Q;K;mFAShB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;mFASd,qB;MAAQ,OAAQ,O;K;+EAEiB,qB;MAAQ,OAAW,U;K;+EAEb,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAQ,Q;K;6EASd,qB;MAAQ,OAAW,U;K;qFAEV,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAQ,Q;K;2FAEb,qB;MAAQ,OAAa,Y;K;6FAEpB,qB;MAAQ,OAAc,a;K;mFAE3B,qB;MAAQ,OAAQ,Q;K;6EAS1B,qB;MAAQ,OAAW,U;K;6EAEEd,qB;MAAQ,OAAW,U;K;qFAEV,qB;MAAQ,OAAU,S;K;+EASjB,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAQ,Q;K;+EASrB,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;iFASjB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc,a;K;qFAE1B,qB;MAAQ,OAAU,S;K;iFASb,qB;MAAQ,OAAO,M;

K;uFAEZ,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAA Y,W;K;+EAE1B,qB;MAAQ,OAAM,K;K;qFAEX,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;+EASrB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAC,a;K;qFAS1B,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;+EASX,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAS,Q;K;iFASnB,qB;MAAQ,OAAO,M;K;qFAEZ,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;kFASJ,qB;MAAQ,OAAQ,O;K;oFAEf,qB;MAAQ,OAAS,Q;K;8EAEpB,qB;MAAQ,OAAM,K;K;oFAEV,qB;MAAQ,OAAU,S;K;mFASzC,qB;MAAQ,OAAS,Q;K;mFAEjB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OA AU,S;K;qFAE1B,qB;MAAQ,OAAU,S;K;wIEx+M7E,wM;MAEiD,qB;QAAA,QAAkB,I;MAAM,sB;QAAA,SAAM B,I;MAAM,2B;QAAA,cAAwB,I;MAAM,yB;QAAA,YAAsB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,a AAuB,I;MAAM,sB;QAAA,SAAM B,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,gC;QAA A,mBAA6B,I;MAAM,+B;QAAA,kBAA4B,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,uB;QAAA,UAAoB,I;MAA M,4B;QAAA,eAAyB,I;MAAM,wB;QAAA,WAAqB,I;MAAM,uB;QAAA,UAAoB,I;MACrf,QAAQ,E;MACR,EA AE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB, S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IAAc,M;MACd,EAAE ,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IA AuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wHAsDX,wM;MAEYc,qB;QAAA,Q AAqB,S;MAAW,sB;QAAA,SAAsB,S;MAAW,2B;QAAA,cAA4B,S;MAAW,yB;QAAA,YAA0B,S;MAAW,0B;Q AAA,aAA6B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,sB;QAAA,SAAuB,S;MAAW,0B;QAAA,aAA0B,S;MAAW ,0B;QAAA,aAA0B,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,+B;QAAA,kBAAmC,S;MAAW,gC;QAAA,mBAA oC,S;MAAW,uB;QAAA,UAAwB,S;MAAW,4B;QAAA,eAA4B,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QA AA,UAAmB,S;MACtnB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aA AF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U ;MACiB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE, kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IA Ae,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OA AO,C;K;sHAYX,kN;MAEwC,wB;QAAA,WAA4C,S;MAAW,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S; MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aA AsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,gC;QAA A,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;M AAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MAC91B,QAAQ,E;M ACR,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF, IAAM B,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;M ACiB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kB AAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IA Ae,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OA AO,C;K;0HAsDX,wM;MAE0C,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;M AAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,sB;QAAA,SAAk B,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAA A,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAA W,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACziB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IA AkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB ,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kB AAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q; MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;gHAyDX,wM;MAEqC,qB;QAAA,QAAc,S;MAAW,sB;QAAA, SAAe,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;Q AAA,aAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAmB,S;MAAW,0B;QAAA,aAAmB,S;MAA W,gC;QAAA,mBAA6B,S;MAAW,+B;QAAA,kBAA4B,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,uB;QAAA,UA

AmB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACxhB,Q  
AAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EA  
AE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IA  
Ac,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACx  
B,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAA  
F,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;8HAqBX,gD;M  
AEsE,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MAC  
R,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAA  
gB,Q;MACHB,OAAO,C;K;sIAoBX,gD;MAEgD,qB;QAAA,QAAiB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;Q  
AAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SA  
AF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wHAWCX,wB;  
MAEyC,qB;QAAA,QAAiB,K;MAAO,qB;QAAA,QAAiB,K;MAC9E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MA  
Cb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;kGAYBX,oB;MAE8B,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MA  
Ae,S;MACnE,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;oHA  
YX,kC;MAEuC,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,mB;QAAA,MAAe,S;MAAW,mB;Q  
AAA,MAAe,S;MACpI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAA  
F,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gGAYX,oB;MAE6B,mB;QAAA,MAAY,S;MAA  
W,mB;QAAA,MAAY,S;MAC5D,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MAC  
X,OAAO,C;K;kHAYX,kC;MAEsC,qB;QAAA,QAAc,S;MAAW,qB;QAAA,QAAc,S;MAAW,mB;QAAA,MAAY,  
S;MAAW,mB;QAAA,MAAY,S;MACvH,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;  
MACb,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gIAeX,wB;MAE6C,qB;QAA  
A,QAaKB,S;MAAW,qB;QAAA,QAaKB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF  
,IAAa,K;MACb,OAAO,C;K;oIAeX,wB;MAE+C,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MACxF,QA  
AQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;4FAKX,Y;MAGI,QAAQ,E  
;MACR,OAAO,C;K;oFAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;8FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C  
;K;kGASX,oB;MAE8B,wB;QAAA,WAAkC,S;MAC5D,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,OAA  
O,C;K;4FAUmE,qB;MAAQ,OAAO,M;K;8FAEd,qB;MAAQ,OAAQ,O;K;4FASrB,qB;MAAQ,OAAO,M;K;0GAE  
R,qB;MAAQ,OAAc,a;K;8FAE7B,qB;MAAQ,OAAO,M;K;gGAEd,qB;MAAQ,OAAQ,O;K;8FASjB,qB;MAAQ,O  
AAO,M;K;gHAEL,qB;MAAQ,OAAiB,gB;K;wGASrC,qB;MAAQ,OAAa,Y;K;0GAEpB,qB;MAAQ,OAAc,a;K;w  
GAEvB,qB;MAAQ,OAAa,Y;K;oFCroB7F,4B;MAE6E,iBAAy,KAAZ,C;K;iGASnB,qB;MAAQ,OAAQ,Q;K;6FAE  
nB,qB;MAAQ,OAAO,M;K;+FAEd,qB;MAAQ,OAAQ,O;K;iGASF,qB;MAAQ,OAAU,S;K;+FAEnB,qB;MAAQ,O  
AAS,Q;K;mGAS3B,qB;MAAQ,OAAW,U;K;mGAEnB,qB;MAAQ,OAAW,U;K;6GC1D/E,mb;MAEmC,yB;QAA  
A,YAAkC,C;MAAG,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAaKB,G;MAAK,wB;QAAA,WAAmB,G;MAAI,k  
C;QAAA,qBAA6B,G;MAAI,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAA  
G,2B;QAAA,cAAuB,E;MAAI,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;  
MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UA  
AkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAaKB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QA  
AA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MA  
AO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBA  
A0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;  
QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAaE,C;MA  
AG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACiB,QAAQ,E;MACR  
,EAAE,WAAF,IAAiB,S;MACjB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAg  
B,Q;MACHB,EAAE,oBAAF,IAA0B,kB;MACiB,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EA  
AE,OAAF,IAAa,K;MACb,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,  
O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IA  
Ac,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE  
,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MA

Cf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GC1BX,0C;MAEwC,oB;QAAA,OAAiB,I;MAAM,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MACpI,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;4EAmIX,4B;MAEKe,iBAAY,KAAZ,C;K;4EAEIE,qC;MAE2E,UAAAY,KAAZ,IAAqB,O;K;4EAIbHG,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,qC;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAIbPg,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,qC;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAIgPg,4B;MAEoE,iBAAY,KAAZ,C;K;2EAEpE,qC;MAE4E,UAAAY,KAAZ,IAAqB,O;K;4EAKcJG,4B;MAE6E,iBAAY,KAAZ,C;K;4EAE7E,qC;MAEqF,UAAAY,KAAZ,IAAqB,O;K;4EAgP1G,4B;MAEqE,iBAAY,KAAZ,C;K;4EAErE,qC;MAE6E,UAAAY,KAAZ,IAAqB,O;K;uFJ57BIG,+H;MAE8B,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,8B;QAAA,iBAA0B,S;MAAW,oB;QAAA,OAAqB,S;MAAW,2B;QAAA,cAAmC,S;MAAW,qB;QAAA,QAAuB,S;MAAW,wB;QAAA,WAA6B,S;MAAW,yB;QAAA,YAAqB,S;MAAW,yB;QAAA,YAAsB,S;MAAW,wB;QAAA,WAAe,S;MAC5Z,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,OAAF,IAAa,K;MACb,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,QAAF,IAAc,Q;MACd,OAAO,C;K;yFA0CX,uC;MAE+B,sB;QAAA,SAAiB,G;MAAK,0B;QAAA,aAAsB,I;MAAM,uB;QAAA,UAAmB,S;MACHG,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;qFAUgD,qB;MAAQ,OAAQ,E;K;mFAEX,qB;MAAQ,OAAQ,O;K;iFAEjB,qB;MAAQ,OAAO,M;K;mFAEd,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;mFAEIB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;qFASF,qB;MAAQ,OAAG,E;K;yFAER,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;qFAEd,qB;MAAQ,OAAQ,O;K;yFAEb,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qFAEjB,qB;MAAQ,OAAS,Q;K;uFAEjB,qB;MAAQ,OAAS,Q;K;mGAEV,qB;MAAQ,OAAgB,e;K;iGAEzB,qB;MAAQ,OAAe,c;K;qFAE9B,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;yFASzB,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;iFAErB,qB;MAAQ,OAAO,M;K;iFASD,qB;MAAQ,OAAO,M;K;iGAER,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAAU,S;K;yFAEjB,qB;MAAQ,OAAW,U;K;qFAErB,qB;MAAQ,OAAS,Q;K;yFAEf,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;qGAEnB,qB;MAAQ,OAAiB,gB;K;qFAS3B,qB;MAAQ,OAAS,Q;K;mFAEIB,qB;MAAQ,OAAQ,O;K;uFAEf,qB;MAAQ,OAAS,Q;K;mFASxB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;yFAEZ,qB;MAAQ,OAAU,S;K;qFAEpB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qGAET,qB;MAAQ,OAAiB,gB;K;+FKnR/F,gB;MAEkC,oB;QAAA,OAAgB,E;MAC9C,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;+FAiBX,8B;MAEkC,4B;QAAA,eAAqB,S;MAAW,oB;QAAA,OAAgB,E;MAC9E,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;0EAUX,4B;MAE6D,iBAAY,KAAZ,C;K;+GC6B7D,sJ;MAEsC,mB;QAAA,MA4GuD,M;MA5GG,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,E;MAAI,mB;QAAA,MAAe,E;MAAI,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,sB;QAAA,SAAmB,K;MAAO,oB;QAAA,OAAa,I;MAAM,uB;QAAA,UAAc,E;MAC/gB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,QAAF,IAAc,M;MACd,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAWX,+B;MAEsE,oB;QAAA,OAAgB,S;MACIF,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAA



E,MAAF,IAAY,I;MACZ,OAAO,C;K;qHASX,e;MAEyC,mB;QAAA,MAAe,E;MACpD,QAAQ,E;MACR,EAAE, KAAF,IAAW,G;MACX,OAAO,C;K;mHAyBX,+D;MAEqE,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;M AAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACrK,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MA CpB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,I AAgB,Q;MACHB,OAAO,C;K;iGAUwE,qB;MAAQ,OAAU,S;K;6FAEnB,qB;MAAQ,OAAS,Q;K;+FAEhB,qB;M AAQ,OAAU,S;K;2FASvB,qB;MAAQ,OAAO,M;K;yFAEhB,qB;MAAQ,OAAM,K;K;yFAEd,qB;MAAQ,OAAM, K;K;yGCrJ3F,uB;MAEsC,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,ORy9MW,S;;MQx9MzE,QAAQ,E;MACR,EA AE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;6HAuCX,mF;MAEgD,oB;QAAA,OAAa,S; MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAA2C,S;MAAW,qB;QAAA,QA A6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/S,QAA Q,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAA E,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MA CIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;uGA2DX,qC;MAEqC,mC;QAAA,sBAAGC,K;MAAO,oB;QAA A,OA4UD,Q;;MA3UvE,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,EAAE,MAAF,IAAY,I;MACZ,OA AO,C;K;yGAmBX,yC;MAEsC,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB, K;MACHH,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB, Q;MACHB,OAAO,C;K;yGAsBX,2B;MAGI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe, O;MACf,OAAO,C;K;+FA8BX,sE;MAEoD,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAqB,K;MAAO,uB;QA AA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpL,QAAQ,E;MACR,EAAE,SA AAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,SAAF,IAAe,O;MA Cf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GAuBX,0D;MAE2D,sB;QAA A,SAAkB,M;MAAQ,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/J, QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE, YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;2GAaX,qC;MAE4D,sB;QAAA,SAAkB,S; MAAW,uB;QAAA,UAAoB,S;MAC/G,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M; MACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;uHAuCX,mF;MAE6C,oB;QAAA,OAAa,S;MAAW,sB;QAAA,S AAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAAmD,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;Q AAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpT,QAAQ,E;MACR,EAAE,M AAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MA Cd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IA AgB,Q;MACHB,OAAO,C;K;qGA+BX,6D;MAEoC,4B;QAAA,eAAyB,K;MAAO,4B;QAAA,eAAyB,K;MAAO,0B ;QAAA,aAAuB,K;MAAO,yB;QAAA,YAAqB,S;MACnJ,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAA E,cAAF,IAAoB,Y;MACpB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,WAAF,IAAiB,S;MACjB,OAAO,C;K;yGakB X,4C;MAEsC,oB;QAAA,OAAGB,S;MAAW,uB;QAAA,UAAoB,S;MAAW,wB;QAAA,WAAb,S;MAAW,uB;QA AA,UAA8B,S;MAC3J,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,EAAE,UAA F,IAAgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FAkCmE,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MA AQ,OAAY,W;K;+FAEnB,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;6FAEpB,qB;MAAQ,OAAY, W;K;6FAStC,qB;MAAQ,OAAY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;uFAEvB,qB;MAAQ,OAAS,Q;K;qFAEn B,qB;MAAQ,OAAO,M;K;uFASX,qB;MAAQ,OAAS,Q;K;yFAEjB,qB;MAAQ,OAAS,Q;K;qGAEX,qB;MAAQ,O AAe,c;K;iFAEhC,qB;MAAQ,OAAM,K;K;iGCharE,0E;MAEoC,gC;QAAA,mBAA6B,K;MAAO,sB;QAAA,SAAk B,C;MAAG,qB;QAAA,QAAiB,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA ,WAAqB,K;MAC3L,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,QAAF,IAAc,M;MACd,EAAE, OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MA ChB,OAAO,C;K;mFAU8E,qB;MAAQ,OAAG,E;K;+FAEL,qB;MAAQ,OAAC,a;K;iFAE7B,qB;MAAQ,OAAO,M; K;yFAEX,qB;MAAQ,OAAW,U;K;+EAEvB,qB;MAAQ,OAAO,M;K;+EAEf,qB;MAAQ,OAAO,M;K;oEtIjIvG,yB ;MAAA,kF;MAAA,0B;MAAA,uB;QaaI,IAAI,OAAO,CAAP,IAA8B,OAAO,KAAzC,C;UACI,MAAM,8BAAyB, wBAAqB,IAA9C,C;;QAEV,OAAY,OAAL,IAAK,C;O;KAhBhB,C;0EawCiC,qB;MAAQ,OAAA,SAAK,I;K;IuInB V,6B;MAAC,qB;QAAA,8C;MAAA,kB;K;IACjC,2C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,yC;MAAA,4C;O;

MAKI,0E;MAEA,sE;K;;IAFA,kD;MAAA,+B;MAAA,0C;K;;IAEA,gD;MAAA,+B;MAAA,wC;K;;IAPJ,qC;MAA  
A,yF;K;;IAAA,0C;MAAA,a;AAAA,S;UAAA,+C;aAAA,O;UAAA,6C;;UAAA,8D;;K;;IA0BmC,sC;MACnC,8B;K;  
;IAMqC,sC;MACrC,8B;K;;IC5DJ,iC;K;;ICMA,4B;K;;IA6BA,gD;K;;IC5BA,qC;K;;IA0BA,+B;K;;ICNqC,uC;MA  
CjC,uB;QAAA,UAAaB,E;MACTb,qB;QAAA,+C;MADA,sB;MACA,kB;K;IAEA,4C;MAAA,e;MAAA,iB;MAAA,  
uB;K;IAAA,0C;MAAA,6C;O;MAKI,4E;MAGA,wE;K;;IAHA,mD;MAAA,gC;MAAA,2C;K;;IAGA,iD;MAAA,gC  
;MAAA,yC;K;;IARJ,sC;MAAA,2F;K;;IAAA,2C;MAAA,a;AAAA,S;UAAA,gD;aAAA,O;UAAA,8C;;UAAA,+D;;  
K;;IAyByB,4B;MACzB,8B;K;;IC/C4C,8B;K;kDAI5C,mB;MAA6D,c;;QrJ2rD7C,Q;QADhB,IAAI,mCAAsB,cAA  
1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,sB;QAAhB,OAAgB,cAAhB,C;UAAgB,2B;UAAM,IqJ3rD6C,OrJ2rD/B,  
SqJ3rD+B,UrJ2rD7C,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;MqJ5rDsD,iB;K;uDAE7D,oB;MACa,c;;Qr  
JmqDG,Q;QADhB,IAAI,cqJlqDA,QrJkqDA,iBqJlqDA,QrJkqDsB,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,O  
qJnqDZ,QrJmqDY,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CqJnqDP,oBrJmqDkB,OqJnqDIB,CrJ  
mqDG,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;MqJpQDH,iB;K;2CAEJ,Y;MAAkC,qBAAQ,C;K;IAEqB,q  
E;MAAA,qB;QAC3D,OAAI,OAAO,uBAAX,GAAiB,mBAAjB,GAA6C,SAAH,EAAG,C;O;K;4CADjD,Y;MAAk  
C,4BAAa,IAAb,EAAMb,GAAnB,EAawB,GAAXb,kBAA6B,wCAA7B,C;K;2CAIIC,Y;MAI4C,uBAAgB,IAAhB,  
C;K;mDAE5C,iB;MAI4D,yBAAgB,IAAhB,EAAsB,KAAtB,C;K;;IC/BhE,8B;MAAA,e;MAAA,iB;MAAA,uB;K;I  
AAA,4B;MAAA,+B;O;MACI,4C;MACA,kD;MACA,0C;MACA,8C;K;;IAHA,mC;MAAA,kB;MAAA,2B;K;;IAC  
A,sC;MAAA,kB;MAAA,8B;K;;IACA,kC;MAAA,kB;MAAA,0B;K;;IACA,oC;MAAA,kB;MAAA,4B;K;;IAJJ,wB;  
MAAA,sH;K;;IAAA,6B;MAAA,a;AAAA,O;UAAA,gC;aAAA,U;UAAA,mC;aAAA,M;UAAA,+B;aAAA,Q;UAAA  
iC;;UAAA,6D;;K;;IAOA,4B;MAKI,mD;MACA,2BAA4B,I;K;yCAE5B,Y;MAEiB,IAAN,I;M5JUX,IAAI,E4JXQ,  
mD5JWR,CAAJ,C;QACI,cAda,qB;QAEb,MAAM,gCAAYB,OAAQ,WAAjC,C;;M4JZC,QAAM,oBAAN,M;aACH,  
M;UAAc,Y;UAAAd,K;aACA,O;UAAe,W;UAAf,K;;UACQ,wC;UAHL,K;;MAAP,W;K;sCAOJ,Y;MAIW,Q;MAHP,  
IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,mD;MAEA,OAAO,2F;K;4DAGX,Y;MACI,iD;MACA,kB;MA  
CA,OAAO,kD;K;+CAeX,iB;MAII,2BAAY,K;MACZ,gD;K;sCAGJ,Y;MAII,+C;K;;ICjDkC,wB;MAoFtC,oC;MAp  
FgE,6B;K;sCAIhE,Y;MAAuC,0C;K;2CAEvC,mB;MAAwD,uB;;QvJkU3C,Q;QADb,YAAY,C;QACC,sB;QAAb,O  
AAa,cAAb,C;UAAa,sB;UACT,IuJnUmE,OvJmUrD,IuJnUqD,UvJmUnE,C;YACI,sBAAO,K;YAAP,wB;;UACJ,qB  
;;QAEJ,sBAAO,E;;MuJvUiD,0B;K;+CAExD,mB;MAA4D,sB;;QvJ2V5D,eAAoB,0BAAa,SAAb,C;QACpB,OAA  
O,QAAS,cAAhB,C;UACI,IuJ7VsE,OvJ6VxD,QAAS,WuJ7V+C,UvJ6VtE,C;YACI,qBAAO,QAAS,Y;YAAhB,uB;  
;;QAGR,qBAAO,E;;MuJjWqD,yB;K;0CAE5D,Y;MAA+C,+CAAiB,CAAjB,C;K;kDAE/C,iB;MAAyD,+CAAiB,  
KAAjB,C;K;6CAEzD,8B;MAA8D,gCAAQ,IAAR,EAAC,SAAd,EAAYB,OAAzB,C;K;IAEIC,wD;MAAgF,uB;MA  
A/E,kB;MAAmC,4B;MAC5D,eAAyB,C;MAGrB,+DAAkB,gBAAIB,EAA6B,OAA7B,EAAsC,WAACK,KAA3C,C;  
MACA,eAAa,UAAU,gBAAV,I;K;iDAGjB,iB;MACI,+DAAkB,KAAIB,EAAYB,YAAzB,C;MAEA,OAAO,wBAA  
K,mBAAY,KAAZ,IAAL,C;K;4FAGY,Y;MAAQ,mB;K;;oCAGnC,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAA  
O,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,2DAAC,IAAd,EAAoB,KAApB,C;K;sCAGX,Y;M  
AG+B,oEAAgB,IAAhB,C;K;IAE/B,2C;MAAA,oB;MACI,eACsB,C;K;kDAEtB,Y;MAAkC,sBAAQ,gB;K;+CAE1  
C,Y;MAEe,gB;MADX,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACX,iE;MAAX,OAAO,+B;K;;IAO0B,sD;M  
AHZC,oB;MAGwD,iD;MAGhD,gEAAmB,KAAAnB,EAA0B,WAAkB,KAA5C,C;MACA,eAAa,K;K;0DAGjB,Y;M  
AAsC,sBAAQ,C;K;wDAE9C,Y;MAAgC,mB;K;uDAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B;  
MAC1B,OAAO,yBAAL,mCAAJ,EAAL,YAAJ,E;K;4DAGX,Y;MAAoC,sBAAQ,CAAR,I;K;;IAGxC,kC;MAAA,sC  
;K;iEACI,uB;MACI,IAAI,QAAQ,CAAR,IAAa,SAAS,IAA1B,C;QACI,MAAM,8BAA0B,YAAS,KAAT,gBAAuB,  
IAAjD,C;;K;kEAIId,uB;MACI,IAAI,QAAQ,CAAR,IAAa,QAAQ,IAAzB,C;QACI,MAAM,8BAA0B,YAAS,KAAT  
gBAAuB,IAAjD,C;;K;iEAIId,oC;MACI,IAAI,YAAY,CAAZ,IAAiB,UAAU,IAA/B,C;QACI,MAAM,8BAA0B,gB  
AAa,SAAb,mBAAkC,OAAIC,gBAakD,IAA5E,C;;MAEV,IAAI,YAAY,OAAhB,C;QACI,MAAM,gCAAyB,gBA  
Aa,SAAb,oBAAmC,OAA5D,C;;K;kEAIId,sC;MACI,IAAI,aAAa,CAAb,IAAkB,WAaw,IAAjC,C;QACI,MAAM,8  
BAA0B,iBAAc,UAAAd,oBAAqC,QAArC,gBAAsD,IAAhF,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAA  
yB,iBAAc,UAAAd,qBAAsC,QAA/D,C;;K;+DAId,a;MAEc,UACsB,M;MAFhC,iBAAc,C;MACL,mB;MAAV,OAA  
U,cAAV,C;QAAU,mB;QACN,aAAW,MAAK,UAAAL,SAAiB,6DAAiB,CAAIC,K;;MAEf,OAAO,U;K;6DAGX,oB  
;MAIiB,Q;MAHb,IAAI,CAAE,KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MAEjC,oBAAoB,KAAM,W;  
MACb,mB;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,gBAAgB,aAAc,O;QAC9B,IAAI,cAAQ,SAAR,CAAJ,C;UAC

I,OAAO,K;;;MAGf,OAAO,I;K;;;IAjDf,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;ICnFwC,uB;MAyHxC,mC;MAz  
CA,uBAC6B,I;MAMc7B,yBACsC,I;K;8CAnHtC,e;MACI,OAAO,6BAAc,GAAd,S;K;gDAGX,iB;MAAwE,gBAA  
R,Y;MAAQ,c;;QxJkrDxD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;QAAhB  
,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IwJlrDwD,OxJkrD1C,OwJlrD6C,MAAH,QxJkrDxD,C;YAAwB,aAAO,I;  
YAAP,e;;QAC9C,aAAO,K;;;MwJnrDyD,iB;K;kDAEhE,iB;MAEI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,UA  
AU,KAAM,I;MACHb,YAAY,KAAM,M;MrKiNO,Q;MqKhNzB,erKgN4C,CAAnB,mDAAmB,YqKhNzB,GrKgN  
yB,C;MqK9M5C,IAAI,eAAS,QAAT,CAAJ,C;QACI,OAAO,K;;MAIP,6B;MAAA,W;QrK0NqB,U;QqK1ND,UrK0  
NoB,CAAnB,uDAAmB,oBqK1NP,GrK0NO,C;;MqK1N5C,W;QACI,OAAO,K;;MAGX,OAAO,I;K;mCAIX,iB;M  
AMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAyB,OAAO,K;MACHc,IAAI,cAAQ,K  
AAM,KAAlB,C;QAAwB,OAAO,K;MAEV,gBAAd,KAAM,Q;MAAQ,c;;QxJ6nDT,Q;QADhB,IAAI,wCAAsB,mB  
AA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CwJ7nDK,2  
BxJ6nDM,OwJ7nDN,CxJ6nDT,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MwJ9nDH,iB;K;sCAGJ,e;MAA  
wC,Q;MAAA,4CAAc,GAAd,8B;K;qCAGxC,Y;MAK+B,OAAQ,SAAR,YAAQ,C;K;oCAEvC,Y;MAAkC,qBAAQ  
,C;K;mFACnB,Y;MAAQ,OAAA,YAAQ,K;K;IAWnB,0E;MAAA,wC;MAAS,sB;K;8EACb,mB;MAAsD,+CAAY,  
OAAZ,C;K;IAI3C,sG;MAAA,kD;K;8FACH,Y;MAAkC,OAAA,0BAAc,U;K;2FACHd,Y;MAAyB,OAAA,0BAAc,  
OAAO,I;K;;wEAItd,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,+F;K;SHAMmB,Y;MAAQ,OAAA,qBAAiB,K;K;;mF  
Ab5D,Y;MACI,IAAI,4BAAJ,C;QACI,+E;;MAcJ,OAAO,mC;K;IAOwD,uD;MAAA,qB;QAAE,2CAAS,EAAT,C;  
O;K;qCAAzE,Y;MAAkC,OAAQ,eAAR,YAAQ,EAAa,IAAb,EAAmB,GAAnB,EAAwB,GAAxB,kBAA6B,iCAA7  
B,C;K;+CAE1C,iB;MAAuD,+BAAS,KAAM,IAAf,IAAsB,GAAtB,GAA4B,wBAAS,KAAM,MAAf,C;K;+CAEnF,  
a;MAAwC,OAAI,MAAM,IAAV,GAAGB,YAAhB,GAAoC,SAAF,CAAE,C;K;IAWtd,4E;MAAA,wC;MAAS,6B;  
K;gFACf,mB;MAAsE,iDAAc,OAAAd,C;K;IAI3D,wG;MAAA,kD;K;gGACH,Y;MAAkC,OAAA,0BAAc,U;K;6FA  
ChD,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;;0EAItd,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,iG;K;wHAMmB,Y  
;MAAQ,OAAA,qBAAiB,K;K;;qFAb5D,Y;MACI,IAAI,8BAAJ,C;QACI,mF;;MAcJ,OAAO,qC;K;oDAMf,e;MAA  
8D,gBAAR,Y;MAAQ,sB;;QxJiJ9C,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IwJjJsD,OxJiJx  
C,OwJjJ2C,IAAH,MxJiJtd,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;MwJlJ+C,yB;K;IAEtD,iC;MA  
AA,qC;K;4DAEI,a;MAAiE,gC;MAAX,OAAU,CAAC,kBAAN,CAAM,0DAAmB,CAApB,KAA4B,oBAAjC,CAA  
iC,8DAAqB,CAAjD,C;K;4DACHe,a;MAAyD,OAAU,SAAL,CAAO,IAAF,mBAAL,CAAY,MAAP,C;K;0DACnE,  
oB;MACI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,OAAO,OAAA,CAAE,IAAF,EAAS,KAAM,IAAf,KAAsB,O  
AAA,CAAE,MAAF,EAAW,KAAM,MAAjB,C;K;;IANrC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;IChIqC,uB;  
MAkBrC,mC;MAIB+D,6B;K;mCAE/D,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,  
C;QAAsB,OAAO,K;MAC7B,OAAO,sDAAU,IAAV,EAAGB,KAAhB,C;K;qCAGX,Y;MAG+B,qEAAkB,IAAIB,C  
;K;IAE/B,iC;MAAA,qC;K;gEACI,a;MAEoB,Q;MADhB,iBAAE,C;MACC,mB;MAAhB,OAAgB,cAAhB,C;QAAg  
B,yB;QACC,U;QAAb,2BAAa,yEAAuB,CAApC,K;;MAEJ,OAAO,U;K;wDAGX,oB;MACI,IAAI,CAAE,KAAP,K  
AAU,KAAM,KAAPB,C;QAA0B,OAAO,K;MACjC,OAAO,CvK40sG,qBuK50xF,KvK40wF,C;K;;IuKvPrH,6C;  
MAAA,4C;QAAA,2B;;MAAA,qC;K;;MCghBA,kC;MA9hBA,cAAwB,C;MACxB,yB;MAEA,sBAAYB,C;;kFAAZ  
B,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;4CA8BA,uB;MAOI,IAAI,cAAc,CAAlB,C;QAAqB,MAAM,6BAAsB,  
mBAAtB,C;MAC3B,IAAI,eAAe,kBAAY,OAA/B,C;QAAqC,M;MACrC,IAAI,uBAAGB,qDAApB,C;QACI,qBAA  
c,gBAAYB,gBAAZ,WAAy,EAAC,EAAd,CAAzB,O;QACd,M;;MAGJ,kBAAkB,uDAAY,kBAAY,OAAxB,EAA8  
B,WAA9B,C;MACIB,oBAAa,WAAb,C;K;0CAGJ,uB;MAII,kBAAkB,gBAAmB,WAAAnB,O;M/J20BtB,U+J10BI,k  
B/J00BJ,E+J10ByB,W/J00BzB,E+J10BsC,C/J00BtC,E+J10ByC,W/J00BzC,E+J10B+C,kBAAY,O/J00B3D,C;MA  
AA,U+Jz0BI,kB/Jy0BJ,E+Jz0ByB,W/Jy0BzB,E+Jz0BsC,kBAAY,OAAZ,GAAMB,WAAAnB,I/Jy0BtC,E+Jz0B+D,  
C/Jy0B/D,E+Jz0BkE,W/Jy0BIE,C;M+Jx0BI,cAAO,C;MACP,qBAAc,W;K;yCAGIb,yB;MAGW,Q;MAAP,OAAO,  
2BAAY,aAAZ,4D;K;yCAGX,iB;MAA2C,OAAI,SAAS,kBAAY,OAAzB,GAA+B,QAAQ,kBAAY,OAApB,IAA/B  
,GAA6D,K;K;yCAExG,iB;MAA2C,OAAI,QAAQ,CAAZ,GAAe,QAAQ,kBAAY,OAApB,IAAf,GAA6C,K;K;2CA  
ExF,iB;MACoD,0BAAY,cAAO,KAAP,IAAZ,C;K;yCAEpD,iB;MAA2C,OAAI,UAAqB,cAAZ,kBAAY,CAAzB,G  
AAoC,CAApC,GAA2C,QAAQ,CAAR,I;K;yCAEtF,iB;MAA2C,OAAI,UAAqB,cAAZ,kBAAY,CAAzB,G  
AA5B,GAA2C,QAAQ,CAAR,I;K;mCAEtF,Y;MAAkC,qBAAQ,C;K;iCAE1C,Y;MAGwB,IAAI,cAAJ,C;QAAe,MA  
AM,2BAAuB,sBAAvB,C;;QAnBIC,Q;QAmBa,OAnBb,2BAmbkG,WAnBIG,4D;;K;uCAqBX,Y;MAG+B,Q;MAA

A,IAAI,cAAJ,C;QAAA,OAAe,I;;QAxBnC,U;QAwBoB,OAxBpB,6BAwByD,WAxBzD,gE;;MAwBoB,W;K;gCA  
E/B,Y;MAGuB,IAAI,cAAJ,C;QA Ae,MAAM,2BAAuB,sBAAvB,C;;QA7BjC,Q;QA6BY,OA7BZ,2BAQyC,mBAA  
Y,cAqB0D,sBArB1D,IAAZ,CARzC,4D;;K;S CA+BX,Y;MAG8B,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAe,I;;QAICl  
C,U;QAKcMB,OAlCnB,6BAQyC,mBAAy,cA0BiB,sBA1BjB,IAAZ,CARzC,gE;;MAkCmB,W;K;0CAE9B,mB;M  
AII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,cAAO,mBAAy,WAAZ,C;MACP,mBAAy,WAAZ,IAAoB,O;MACpB,  
wBAAQ,CAAR,I;K;yCAGJ,mB;MAII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,mBA7CgD,mBAAy,cA6CIC,SA7C  
kC,IAAZ,CA6ChD,IAAmC,O;MACnC,wBAAQ,CAAR,I;K;uCAGJ,Y;MAII,IAAI,cAAJ,C;QA Ae,MAAM,2BAAu  
B,sBAAvB,C;MA7Dd,Q;MA+DP,cA/DO,2BA+DmB,WA/DnB,4D;MAGeP,mBAAy,WAAZ,IAAoB,I;MACpB,c  
AAO,mBAAy,WAAZ,C;MACP,wBAAQ,CAAR,I;MACA,OAAO,O;K;6CAGX,Y;MAGqC,OAAI,cAAJ,GAAe,I  
AAf,GAAyB,kB;K;sCAE9D,Y;MAII,IAAI,cAAJ,C;QA Ae,MAAM,2BAAuB,sBAAvB,C;MAErB,wBAzEgD,mBA  
AY,cAyEtB,sBAzEsB,IAAZ,C;MARzC,Q;MAkFP,cAlFO,2BAkFmB,iBAIFnB,4D;MAmFP,mBAAy,iBAAZ,IAA  
iC,I;MACjC,wBAAQ,CAAR,I;MACA,OAAO,O;K;4CAGX,Y;MAGoC,OAAI,cAAJ,GAAe,IAAf,GAAyB,iB;K;q  
CAE7D,mB;MAEI,mBAAQ,OAAR,C;MACA,OAAO,I;K;uCAGX,0B;MACI,oCAAa,4BAAmB,KAA nB,EAA0B,  
SAA1B,C;MAEb,IAAI,UAAS,SAAb,C;QACI,mBAAQ,OAAR,C;QACA,M;aACG,IAAI,UAAS,CAAb,C;QACH,o  
BAAS,OAAT,C;QACA,M;;MAGJ,sBA Ae,YAAO,CAAP,IAAf,C;MA2BA,oBAjIgD,mBAAy,cAiI1B,KAjI0B,IA  
AZ,C;MAmIhD,IAAI,QAAS,SAAD,GAAQ,CAAR,IA Ae,CAA3B,C;QAEI,+BAA+B,mBAAy,aAAZ,C;QAC/B,s  
BAAsB,mBAAy,WAAZ,C;QAEtB,IAAI,4BAA4B,WAAhC,C;UACI,mBAAy,eAAZ,IAA+B,mBAAy,WAAZ,C;  
U/JgrB3C,U+J/qBY,kB/J+qBZ,E+J/qBiC,kB/J+qBjC,E+J/qB8C,W/J+qB9C,E+J/qBoD,cAAO,CAAP,I/J+qBpD,E+  
J/qB8D,2BAA2B,CAA3B,I/J+qB9D,C;;UAAA,U+J7qBY,kB/J6qBZ,E+J7qBiC,kB/J6qBjC,E+J7qB8C,cAAO,CA  
AP,I/J6qB9C,E+J7qBwD,W/J6qBxD,E+J7qB8D,kBAAy,O/J6qB1E,C;U+J5qBY,mBAAy,kBAAy,OAAZ,GAAM  
B,CAAnB,IAAZ,IAAoC,mBAAy,CAAZ,C;U/J4qBhD,U+J3qBY,kB/J2qBZ,E+J3qBiC,kB/J2qBjC,E+J3qB8C,C/J  
2qB9C,E+J3qBiD,C/J2qBjD,E+J3qBoD,2BAA2B,CAA3B,I/J2qBpD,C;;Q+JxqBQ,mBAAy,wBAAZ,IAAwC,O;Q  
ACxC,cAAO,e;;QAGP,WArJ4C,mBAAy,cAqJ/B,SArJ+B,IAAZ,C;QAuJ5C,IAAI,gBAAGB,IAApB,C;U/JkqBR,U  
+JjqBY,kB/JiqBZ,E+JjqBiC,kB/JiqBjC,E+JjqB8C,gBAAGB,CAAhB,I/JiqB9C,E+JjqBiE,a/JiqBjE,E+JjqBgF,I/JiqB  
hF,C;;UAAA,U+J/pBY,kB/J+pBZ,E+J/pBiC,kB/J+pBjC,E+J/pB8C,C/J+pB9C,E+J/pBiD,C/J+pBjD,E+J/pBoD,I/J+  
pBpD,C;U+J9pBY,mBAAy,CAAZ,IAAiB,mBAAy,kBAAy,OAAZ,GAAMB,CAAnB,IAAZ,C;U/J8pB7B,U+J7p  
BY,kB/J6pBZ,E+J7pBiC,kB/J6pBjC,E+J7pB8C,gBAAGB,CAAhB,I/J6pB9C,E+J7pBiE,a/J6pBjE,E+J7pBgF,kBA  
AY,OAAZ,GAAMB,CAAnB,I/J6pBhF,C;;Q+J1pBQ,mBAAy,aAAZ,IAA6B,O;;MAEjC,wBAAQ,CAAR,I;K;oDA  
GJ,mC;MAGkD,UAIxB,M;MANtB,eAAe,QAAS,W;MAEsB,OAAZ,kBAAy,O;MAA9C,iBAAc,aAA d,wB;QACI,  
IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAy,KAAZ,IAAqB,QAAS,O;;MAEZ,oB;MAAtB,mBAAc,  
CAAd,8B;QACI,IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAy,OAAZ,IAAqB,QAAS,O;;MAGIc,wB  
AAQ,QAAS,KAAjB,I;K;0CAGJ,oB;MACI,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAC/B,sBA Ae,IAAK,KAA  
L,GAAy,QAAS,KAArB,IAAf,C;MACA,8BA tLgD,mBAAy,cAsLvB,SAtLuB,IAAZ,CAsLhD,EAA4C,QAA5C,C;  
MACA,OAAO,I;K;0CAGX,2B;MACI,oCAAa,4BAAmB,KAA nB,EAA0B,SAA1B,C;MAEb,IAAI,QAAS,UAAb,  
C;QACI,OAAO,K;aACJ,IAAI,UAAS,SAAb,C;QACH,OAAO,oBAAO,QAAP,C;;MAGX,sBA Ae,IAAK,KAA L,G  
AAy,QAAS,KAArB,IAAf,C;MAEA,WArMgD,mBAAy,cAqMnC,SArMmC,IAAZ,C;MA sMhD,oBA tMgD,mBA  
AY,cAsM1B,KAtM0B,IAAZ,C;MAuMhD,mBAAmB,QAAS,K;MAE5B,IAAI,QAAS,SAAD,GAAQ,CAAR,IA Ae,  
CAA3B,C;QAGI,kBAAkB,cAAO,YAAP,I;QAEIb,IAAI,iBAAiB,WAArB,C;UACI,IAAI,eAAe,CAAnB,C;Y/J0m  
BZ,U+JzmBgB,kB/JymBhB,E+JzmBqC,kB/JymBrC,E+JzmBkD,W/JymBiD,E+JzmB+D,W/JymB/D,E+JzmBqE,a/  
JymBrE,C;;Y+JvmBgB,4BA Ae,kBAAy,OAA3B,I;YACA,sBAAsB,gBAAGB,WAAhB,I;YAcTB,kBAAkB,kBAA  
Y,OAAZ,GAAMB,WAA nB,I;YAEIb,IAAI,eAAe,eAA nB,C;c/JmmBhB,U+JlmBoB,kB/JkmBpB,E+JlmByC,kB/Jk  
mBzC,E+JlmBsD,W/JkmBtD,E+JlmBmE,W/JkmBnE,E+JlmByE,a/JkmBzE,C;;cAAA,U+JhmBoB,kB/JgmBpB,E+  
JhmByC,kB/JgmBzC,E+JhmBsD,W/JgmBtD,E+JhmBmE,W/JgmBnE,E+JhmByE,cAAO,WAAP,I/JgmBzE,C;cAA  
A,U+J/lBoB,kB/J+lBpB,E+J/lByC,kB/J+lBzC,E+J/lBsD,C/J+lBtD,E+J/lByD,cAAO,WAAP,I/J+lBzD,E+J/lB6E,a/J  
+lB7E,C;;;UAAA,U+J3lBY,kB/J2lBZ,E+J3lBiC,kB/J2lBjC,E+J3lB8C,W/J2lB9C,E+J3lB2D,W/J2lB3D,E+J3lBiE,  
kBAAy,O/J2lB7E,C;U+J1lBY,IAAI,gBAAGB,aAApB,C;Y/J0lBZ,U+JzlBgB,kB/JylBhB,E+JzlBqC,kB/JylBrC,E+J  
zIbKd,kBAAy,OAAZ,GAAMB,YAA nB,I/JylBiD,E+JzlBmF,C/JylBnF,E+JzlBsF,a/JylBtF,C;;YAAA,U+JvlBgB,k  
B/JulBhB,E+JvlBqC,kB/JulBrC,E+JvlBkD,kBAAy,OAAZ,GAAMB,YAA nB,I/JulBiD,E+JvlBmF,C/JulBnF,E+Jvl

BsF, Y/JulBtF, C; YAAA, U+JtlBgB, kB/JslBhB, E+JtlBqC, kB/JslBrC, E+JtlBkD, C/JslBID, E+JtlBqD, Y/JslBrD, E+JtlBmE, a/JslBnE, C;;; Q+JnlBQ, cAAO, W; QACP, 8BAAuB, mBAAy, gBAAGb, YAAhB, IAAZ, CAAvB, EAAKE, QAAIE, C;;; QAIA, 2BAA2B, gBAAGb, YAAhB, I; QAE3B, IAAI, gBAAGb, IAApB, C; UACI, IAAI, QAAO, YAAP, SAAuB, kBAAY, OAAvC, C; Y/J2kBZ, U+J1kBgB, kB/J0kBhB, E+J1kBqC, kB/J0kBrC, E+J1kBkD, oB/J0kBid, E+J1kBwE, a/J0kBxE, E+J1kBuF, I/J0kBvF, C;;; Y+JxkBgB, IAAI, wBAAwB, kBAAY, OAAx, C; c/JwkBhB, U+JvkBoB, kB/JukBpB, E+JvkByC, kB/JukBzC, E+JvkBsD, uBAAuB, kBAAY, OAAnc, I/JukBtD, E+JvkB+F, a/JukB/F, E+JvkB8G, I/JukB9G, C;;; c+JrkBoB, mBAAmB, OAAO, YAAP, GAASb, kBAAY, OAAIC, I; c/JqkBvC, U+JpkBoB, kB/JokBpB, E+JpkByC, kB/JokBzC, E+JpkBsD, C/JokBtD, E+JpkByD, OAAO, YAAP, I/JokBzD, E+JpkB8E, I/JokB9E, C; cAAA, U+JnkBoB, kB/JmkBpB, E+JnkByC, kB/JmkBzC, E+JnkBsD, oB/JmkBtD, E+JnkB4E, a/JmkB5E, E+JnkB2F, OAAO, YAAP, I/JmkB3F, C;;; UAAA, U+J/jBY, kB/J+jBZ, E+J/jBiC, kB/J+jBjC, E+J/jB8C, Y/J+jB9C, E+J/jB4D, C/J+jB5D, E+J/jB+D, I/J+jB/D, C; U+J9jBY, IAAI, wBAAwB, kBAAY, OAAx, C; Y/J8jBZ, U+J7jBgB, kB/J6jBhB, E+J7jBqC, kB/J6jBrC, E+J7jBkD, uBAuB, kBAAY, OAAnc, I/J6jBID, E+J7jB2F, a/J6jB3F, E+J7jB0G, kBAAY, O/J6jBtH, C;;; YAAA, U+J3jBgB, kB/J2jBhB, E+J3jBqC, kB/J2jBrC, E+J3jBkD, C/J2jBID, E+J3jBqD, kBAAY, OAAZ, GAAMb, YAAAnB, I/J2jBrD, E+J3jBsF, kBAAY, O/J2jBIG, C; YAAA, U+J1jBgB, kB/J0jBhB, E+J1jBqC, kB/J0jBrC, E+J1jBkD, oB/J0jBID, E+J1jBwE, a/J0jBxE, E+J1jBuF, kBAAY, OAAZ, GAAMb, YAAAnB, I/J0jBvF, C;;; Q+JvjBQ, 8BAAuB, aAAvB, EAAsC, QAAc, C;;; MAGJ, OAAO, I; K; uCAGX, iB; MACI, oCAAa, 2BAAk, KAAIB, EAAyB, SAAzB, C; MAjRN, Q; MAmRP, OAnRO, 2BAQyC, mBAAy, cA2Q3B, KA3Q2B, IAAZ, CARzC, 4D; K; uCAsRX, 0B; MACI, oCAAa, 2BAAk, KAAIB, EAAyB, SAAzB, C; MAEb, oBAjRgD, mBAAy, cAiR1B, KAjR0B, IAAZ, C; MARzC, Q; MA0RP, iBA1RO, 2BA0RsB, aA1RtB, 4D; MA2RP, mBAAy, aAAZ, IAA6B, O; MAE7B, OAAO, U; K; OCAGX, mB; MAAoD, 0BAAQ, OAAr, MAAoB, E; K; yCAExE, mB; MAIsB, IAIA, IAJA, EAiuB, M; MAPzC, WA3RgD, mBAAy, cA2RnC, SA3RmC, IAAZ, C; MA6RhD, IAAI, cAAO, IAAx, C; QACI, iBAAc, WAAd, UAAyB, IAAzB, U; UACI, IAAI, gBAAW, mBAAy, KAAZ, CAAX, CAAJ, C; YAAmC, OAAO, QAAQ, WAAR, I;;; aAE3C, IAAI, eAAQ, IAAZ, C; QACW, kB; QAAuB, SAAZ, kBAAY, O; QAArC, qD; UACI, IAAI, gBAAW, mBAAy, OAAZ, CAAX, CAAJ, C; YAAmC, OAAO, UAAQ, WAAR, I;;; QAE9C, mBAAC, CAAd, YAASb, IAAtB, Y; UACI, IAAI, gBAAW, mBAAy, OAAZ, CAAX, CAAJ, C; YAAmC, OAAO, UAAQ, kBAAY, OAApB, GAA2B, WAA3B, I;;; MAIID, OAAO, E; K; 6CAGX, mB; MAIsC, UAOJ, MAPI, EA0a, M; MAV/C, WA9SgD, mBAAy, cA8SnC, SA9SmC, IAAZ, C; MAgThD, IAAI, cAAO, IAAX, C; QACkC, kB; QAA9B, iBAAc, OAAO, CAAP, IAAd, yB; UACI, IAAI, gBAAW, mBAAy, KAAZ, CAAX, CAAJ, C; YAAmC, OAAO, QAAQ, WAAR, I;;; aAE3C, IAAI, cAAO, IAAX, C; QACH, mBAAC, OAAO, CAAP, IAAd, aAA8B, CAA9B, Y; UACI, IAAI, gBAAW, mBAAy, OAAZ, CAAX, CAAJ, C; YAAmC, OAAO, UAAQ, kBAAY, OAApB, GAA2B, WAA3B, I;;; QAEpB, uBAAZ, kBAAY, C; QAAiB, oB; QAA3C, wD; UACI, IAAI, gBAAW, mBAAy, OAAZ, CAAX, CAAJ, C; YAAmC, OAAO, UAAQ, WAAR, I;;; MAIID, OAAO, E; K; wCAGX, mB; MACI, YAAy, mBAAQ, OAAr, C; MACZ, IAAI, UAAS, EAAb, C; QAAiB, OAAO, K; MACxB, sBAAS, KAAT, C; MACA, OAAO, I; K; 4CAGX, iB; MACI, oCAAa, 2BAAk, KAAIB, EAAyB, SAAzB, C; MAEb, IAAI, UAAS, sBAAb, C; QACI, OAAO, iB; aACJ, IAAI, UAAS, CAAb, C; QACH, OAAO, kB;;; MAGX, oBAhVgD, mBAAy, cAgV1B, KAhV0B, IAAZ, C; MARzC, Q; MAyVP, cAzVO, 2BAyVmB, aAzVnB, 4D; MA2VP, IAAI, QAAQ, aAAS, CAARb, C; QAEI, IAAI, iBAAiB, WAARb, C; U/Joer, U+JneY, kB/JmeZ, E+JneiC, kB/JmejC, E+Jne8C, cAAO, CAAP, I/Jme9C, E+JnewD, W/JmexD, E+Jne8D, a/Jme9D, C;;; UAAA, U+JjeY, kB/JjeZ, E+JjeiC, kB/JjeJ, E+Jje8C, C/Jje9C, E+JjeiD, C/JjeJ, E+JjeoD, a/JjepD, C; U+JheY, mBAAy, CAAZ, IAAiB, mBAAy, kBAAY, OAAZ, GAAMb, CAAnB, IAAZ, C; U/Jge7B, U+JdY, kB/J+dZ, E+J/diC, kB/J+djC, E+J/d8C, cAAO, CAAP, I/J+d9C, E+J/dwD, W/J+dxD, E+J/d8D, kBAAY, OAAZ, GAAMb, CAAnB, I/J+d9D, C;;; Q+J5dQ, mBAAy, WAAZ, IAAoB, I; QACPb, cAAO, mBAAy, WAAZ, C;;; QAGP, wBAjW4C, mBAAy, cAiW1B, sBAjWkB, IAAZ, C; QAmW5C, IAAI, iBAAiB, iBAArB, C; U/JsdR, U+JrdY, kB/JqdZ, E+JrdiC, kB/JqdjC, E+Jrd8C, a/Jqd9C, E+Jrd6D, gBAAGb, CAAhB, I/Jqd7D, E+JrdgF, oBAAoB, CAAPb, I/JqdhF, C;;; UAAA, U+JndY, kB/JmdZ, E+JndiC, kB/JmdjC, E+Jnd8C, a/Jmd9C, E+Jnd6D, gBAAGb, CAAhB, I/Jmd7D, E+JndgF, kBAAY, O/Jmd5F, C; U+JldY, mBAAy, kBAAY, OAAZ, GAAMb, CAAnB, IAAZ, IAAoC, mBAAy, CAAZ, C; U/JkdhD, U+JjdY, kB/JidZ, E+JjdiC, kB/JidjC, E+Jjd8C, C/Jid9C, E+JjdiD, C/JidjD, E+JjdoD, oBAAoB, CAAPb, I/JidpD, C;;; Q+J9cQ, mBAAy, iBAAZ, IAAiC, I;;; MAErC, wBAAQ, CAAR, I; MAEA, OAAO, O; K; 6CAGX, oB; MAAkE, 0B;;; QAa5C, wD; QART, aAAL, IAAK, U; QAAL, Y; UAA8B, SAAZ, kB9K6wOnB, YAAQ, C;;; Q8K7wOX, W; UACI, yBAAO, K; UAAP, 2B;;; QAEJ, WA1XgD, mBAAy, cA0XnC, SA1XmC, IAAZ, C; QA2XhD, cAAc, W; QACd, eAAe, K; QAEf, IAAI, cAAO, IAAX, C; UACI, iBAAc, WAAd, UAAyB, IAAzB, U; YACI, cAAc, mBAAy, KAAZ, C; YAGd, IAjBsE, CAAU, wBAiBIE, 0EAjBkE, CAi

BhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAC,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA/BsE,CAAU,wBA+BIE,kFA/BkE,CA+BhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA5CsE,CAAU,wBA4CIE,kFA5CkE,CA4ChF,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;MAvDuD,6B;K;6CAEIE,oB;MAAkE,0B;;QAW5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB9K6wOnB,YAAQ,C;;Q8K7wOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAC,WAAAd,UAAyB,IAAZB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAf+E,wBAeJ,E,0EafiE,CAe/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAC,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA7B+E,wBA6BjE,kFA7BiE,CA6B/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA1C+E,wBA0CjE,kFA1CiE,CA0C/E,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;MArDuD,6B;K;2CAEIE,qB;MASsB,IAII,IAJJ,EAKM,MALN,EAaA,MAbA,EAauB,MAbvB,EAkBI,MAIBJ,EAmBM,MANBN,EA+BI,M;MAvCb,aAAL,IAAK,U;MAAL,Y;QAA8B,SAAZ,kB9K6wOnB,YAAQ,C;;M8K7wOX,W;QACI,OAAO,K;MAEX,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;MA2XhD,cAAc,W;MACd,eAAe,K;MAEf,IAAI,cAAO,IAAX,C;QACI,iBAAC,WAAAd,UAAyB,IAAZB,U;UACI,cAAc,mBAAY,KAAZ,C;UAGd,IAAI,UAAU,0EAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;YAEzB,WAAW,I;;QAGP,OAAZ,kBAAY,EAAC,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;QAGE,oB;QAAuB,SAAZ,kBAAY,O;QAArC,uD;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;YAEzB,WAAW,I;;QAGnB,UAAU,mBAAY,OAAZ,C;QAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,OAAZ,IAAuB,S;YACvB,UAAU,mBAAY,OAAZ,C;;YAEV,WAAW,I;;;MAIvB,IAAI,QAAJ,C;QACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;MAEX,OAAO,Q;K;iCAGX,Y;MACI,WA7agD,mBAAY,cA6anC,SA7amC,IAAZ,C;MA8ahD,IAAI,cAAO,IAAX,C;QACgB,OAAZ,kBAAY,EAAC,IAAL,EAAW,WAAAX,EAaiB,IAAjB,C;;QACT,IxKtS6C,CAAC,cwKsS9C,C;UACS,OAAZ,kBAAY,EAAC,IAAL,EAAW,WAAAX,EAaiB,kBAAY,OAA7B,C;UACA,OAAZ,kBAAY,EAAC,IAAL,EAAW,CAAX,EAAC,IAAd,C;;MAEHb,cAAO,C;MACP,YAAO,C;K;2CAGX,iB;MAGe,IAAC,IAAD,EAAC,M;MAfP,WACW,eAAC,OAAI,KAAM,OAAAN,IAAc,SAAlB,GAAwB,KAAxB,GAAmC,aAAa,KAAb,EAAoB,SAAPB,CAAPC,uB;MAEX,WA7bgD,mBAAY,cA6bnC,SA7bmC,IAAZ,C;MA8bhD,IAAI,cAAO,IAAX,C;Q/J2XJ,U+J1XQ,kB/J0XR,E+J1X6B,I/J0X7B,EAD+F,CAC/F,E+J1XgD,W/J0XhD,E+J1XiE,I/J0XjE,C;;Q+JzXW,IxKtT6C,CAAC,cwKsT9C,C;U/JyXX,U+JxXQ,kB/JwXR,E+JxX6B,I/JwX7B,E+JxXuD,C/JwXvD,E+JxXuE,W/JwXvE,E+JxXwF,kBAAY,O/JwXpG,C;UAAA,U+JvXQ,kB/JuXR,E+JvX6B,I/JuX7B,E+JvXuD,kBAAY,OAAZ,GAAmB,WAAAnB,I/JuXvD,E+JvX6F,C/JuX7F,E+JvX2G,I/JuX3G,C;;M+JrXI,IAAI,IAAK,OAAL,GAAY,SAAhB,C;QACI,KAAK,SAAL,IAAa,I;;MAIjB,OAAO,qD;K;mCAGX,Y;MAEI,OAAO,qBAAQ,gBAAmB,SAAnB,OAAR,C;K;+CAGX,iB;MAC0D,4BAAQ,KAAR,C;K;+CAC1D,Y;MAA0C,qB;K;IAE1C,gC;MAAA,oC;MACI,0BrHriBuC,E;MqHsiBvC,sBAAiC,U;MACjC,4BAAuC,E;K;yDAEvC,oC;MAEI,kBAAkB,eAAe,eAAgB,CAA/B,K;MACIB,IAAI,eAAc,WAAAd,QAA4B,CAAhC,C;QACI,cAAc,W;MACIB,IAAI,eAAc,UAAAd,QAA6B,CAAjC,C;QACI,cAAkB,cAAc,UAAIB,GAAgC,UAAhC,GAAmD,U;MACrE,OAAO,W;K;;IAZf,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;qDagBa,qB;MAEI,WAVEgD,mBAAY,cAuenC,SAvemC,IAAZ,C;MAwehD,WAAe,kBAAa,cAAO,IAAxB,GAA8B,WAA9B,GAAwC,cAAO,kBAAY,OAAAnB,I;MACnD,UAAU,IAAV,EAAGb,cAAhB,C;K;;IA5iBJ,iD;MAAA,oD;MAGwC,+B;MApB5C,sB;MAqBsB,Q;MACV,wBAAmB,CAAnB,C;QAAwB,4D;WACxB,sBAAkB,CAAlB,C;QAAuB,uBAAa,eAAb,O;;QACf,MAAM,gCAAyB,uBAAoB,eAA7C,C;MAHIB,0B;MAJJ,Y;K;IAWA,kC;MAAA,oD;MAGoB,+B;MA/BxB,sB;MAGCQ,sBAAC,qD;MAIJB,Y;K;IAOA,4C;MAAAA,oD;MAG2C,+B;MAtC/C,sB;MAuCQ,sBzJrB8D,YyJqBhD,QzJrBgD,C;MyJsB9D,aAAO,mBAAY,O;MACn

B,IAAI,mB9K+qPD,YAAQ,C8K/qPX,C;QAA2B,sBAAc,qD;MAN7C,Y;K;IC5BJ,4B;MAMoB,Q;M/KghqBA,U;  
MADhB,UAAe,C;MACf,uD;QAAgB,cAAhB,iB;QACI,YAAgB,O+KlhqBiB,O/KkhqBjC,I;M+KlhqBJ,aAAa,iB/K  
ohqBN,G+KphqBM,C;MACb,wBAAgB,SAAhB,gB;QAAgB,gBAAA,SAAhB,M;QACW,SAAP,MAAO,EAAO,S  
AAP,C;MAEX,OAAO,M;K;IAGX,0B;MASiB,Q;MAFb,YAAY,iBAAa,gBAAb,C;MACZ,YAAY,iBAAa,gBAAb,  
C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAA  
I,IAAK,OAAT,C;MAEV,OAAO,UAAAS,KAAT,C;K;gGAGX,qB;MAWW,4B;MAAA,U;QAAqB,OAAL,S/K0qPh  
B,YAAQ,C;M+K1qPf,W;K;oFAGJ,mC;MAUI,O/K6pPO,qBAAQ,C+K7pPf,GAAe,cAAf,GAAMC,S;K;IAGvC,iD  
;MAMI,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAL,KAAa,KA  
AM,OAAxD,C;QAA8D,OAAO,K;MAErE,4C;QACI,SAAS,UAAK,CAAL,C;QACT,SAAS,MAAM,CAAN,C;QA  
ET,IAAI,OAAO,EAAx,C;UACI,Q;eACG,IAAI,cAAc,UAAIB,C;UACH,OAAO,K;QAIP,0BAAsB,kBAAtB,C;U  
AA4C,IAAI,CAAI,kBAAH,EAAG,EAaKb,EAAIB,CAAR,C;YAA+B,OAAO,K;eACIF,8BAAsB,sBAAtB,C;UAA  
4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IA  
AI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,6BAAsB,qBAAtB,C;UAA4C,IAAI,CA  
AI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAA  
H,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EA  
AG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,gCAAsB,wBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EA  
Ac,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EA  
Ad,CAAR,C;YAA2B,OAAO,K;eAC9E,iCAAsB,yBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CA  
AR,C;YAA2B,OAAO,K;eAE9E,qCAAsB,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C  
;YAA2B,OAAO,K;eAC9E,sCAAsB,8BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA  
2B,OAAO,K;eAC9E,oCAAsB,4BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,O  
AAO,K;eAC9E,qCAAsB,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,  
K;eAEtE,IAAI,YAAM,EAAN,CAAJ,C;UAAc,OAAO,K;MAIrC,OAAO,I;K;IAGX,4C;MAKI,IAAI,iBAAJ,C;QA  
AkB,OAAO,M;MACzB,aAAa,CAAK,eAAL,gBAAK,EAaA,SAAb,CAAL,GAA6C,CAA7C,QAAiD,CAAjD,I;Mv  
C6SkB,kBAAxB,mBuC5SY,MvC4SZ,C;MuC3SH,oDzK5BgD,gByK4BhD,C;MADJ,O9JnCO,WuH+U6C,W;K;Iu  
CvSxD,mE;MAEI,IAAY,SAAR,0BAAJ,C;QACI,MAAO,gBAAO,OAAP,C;QACP,M;MAEJ,SAAU,WAAI,SAAJ  
,C;MACV,MAAO,gBAAO,EAAP,C;MAEP,4C;QACI,IAAI,MAAK,CAAT,C;UACI,MAAO,gBAAO,IAAP,C;QA  
EX,cAAc,UAAK,CAAL,C;QAEV,IADE,OACF,S;UAAmB,MAAO,gBAAO,MAAP,C;aAC1B,mBAFE,OAeF,E;U  
AA2B,4BAAR,OAAQ,EAA4B,MAA5B,EAAoC,SAAP,C;aAC3B,uBAHE,OAGF,E;UAAmB,MAAO,gBA Ae,g  
BAAR,OAAQ,CAAF,C;aAC1B,wBAJE,OAIF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,sBAL  
E,OAKF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,uBANE,OAMF,E;UAAmB,MAAO,gBA Ae,  
gBAAR,OAAQ,CAAF,C;aAC1B,wBAPE,OAOF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,yB  
ARE,OAQF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,uBATE,OASF,E;UAAmB,MAAO,gBA  
Ae,gBAAR,OAAQ,CAAF,C;aAC1B,0BAVE,OAUF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aE1B,k  
BAZE,OAYF,c;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBAeE,OAaf,e;UAAmB,MAAO,gBA  
Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBAde,OAcF,a;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,k  
BAFe,OAef,c;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;UAEP,MAAO,gBAAO,OAAQ,WAAf,C;MAII  
C,MAAO,gBAAO,EAAP,C;MACP,SAAU,kBAAmB,iBAAV,SAAU,CAAnB,C;K;ICpJd,uC;MAIqD,+CAAwC,iB  
AAO,CAA/C,IAAoD,mC;K;IAEzG,4D;MAWQ,kBADE,SACF,O;QADJ,OACc,S;WACV,kBAFE,SAEF,c;QAEQ,  
yCAAwB,MAAO,KAAP,GAAc,CAAT,C;UAJZ,OAIuD,S;UAJvD,OAK6B,mBAAL,SAAK,CAAT,GAA+B,sBA  
A/B,GAAgD,S;QALpE,OAogB,oCAAJ,GAA0C,sBAA1C,GAA2D,mB;K;IAG3E,gD;MAWQ,kBADE,SACF,O;  
QADJ,OACc,S;WACV,kBAFE,SAEF,c;QAFJ,OAe8B,mBAAL,SAAK,CAAT,GAA+B,sBAA/B,GAAgD,S;QAFr  
E,OAGgB,oCAAJ,GAA0C,sBAA1C,GAA2D,mB;K;IAG3E,kD;MAKI,OAai,oCAAJ,GAA0C,sBAA1C,GAA2D,o  
B;K;IAE/D,kD;MAKI,OAai,oCAAJ,GAA0C,oBAA1C,GAA2D,iB;K;IIKnD/D,yB;MAAA,6B;K;sCACI,Y;MAA  
kC,Y;K;0CACIC,Y;MAAsC,Y;K;wCACtC,Y;MAAgC,Q;K;4CACHC,Y;MAAoC,S;K;mCACpC,Y;MAA+B,MAA  
M,6B;K;uCACrC,Y;MAAmC,MAAM,6B;K;;IAN7C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;IASA,qB;MAAA,  
yB;MACI,+C;K;ICA EA,iB;MAA4C,qCAAoB,KAAM,U;K;mCACtE,Y;MAA+B,Q;K;mCAC/B,Y;MAAkC,W;K;i  
FAEX,Y;MAAQ,Q;K;kCAC/B,Y;MAAkC,W;K;yCACIC,mB;MAAmD,Y;K;8CACnD,oB;MAAmE,OAAA,QAAS

,U;K;sCAE5E,iB;MAAwC,MAAM,8BAA0B,iDAA8C,KAA9C,MAA1B,C;K;wCAC9C,mB;MAA8C,S;K;4CAC9C,mB;MAAkD,S;K;mCAEID,Y;MAA6C,kC;K;uCAC7C,Y;MAAqD,kC;K;+CACrD,iB;MACl,IAAI,UAAS,CAAb,C;QAAgB,MAAM,8BAA0B,YAAS,KAAnc,C;MACtB,OAAO,2B;K;0CAGX,8B;MACl,IAAI,cAAa,CAAb,IAAkB,YAAW,CAAjC,C;QAAoC,OAAO,I;MAC3C,MAAM,8BAA0B,gBAAa,SAAb,mBAaKc,OAA5D,C;K;wCAGV,Y;MAAiC,8B;K;;IA5BrC,iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IA+BA,iC;MAA8D,6BAAkB,SAaIB,EAAoC,KAApC,C;K;IAE5B,8C;MAAC,oB;MAA0B,0B;K;yFAClC,Y;MAAQ,OAAA,WAAO,O;K;0CACtC,Y;MAAkC,OAAA,WNqqP3B,YAAQ,C;K;iDMpqPf,mB;MAA6C,OAAO,SAAP,WAAO,EAAS,OAAT,C;K;sDACpD,oB;MAAsE,c;;Qc4nDtD,Q;QADhB,IAAI,cd3nDyD,Qc2nDzD,iBd3nDyD,Qc2nDnC,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,Od5nD6C,Qc4nD7C,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,Cd5nDkD,oBc4nDvC,Od5nDuC,Cc4nDtD,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;Md7nDsD,iB;K;2CAC7D,Y;MAAuC,OAAO,qBAAP,WAAO,C;K;0CAC9C,Y;MAC+C,gBAAP,W;MAAA,OAAwB,cAAxB,GegKpC,SfhKoC,GekKpC,SN83BoB,Q;K;;IT7hC5B,qB;MAIsC,8B;K;IAEtC,4B;MAIqD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,OAAT,QAAS,CAAhC,GAA8C,W;K;mFAEnG,yB;MAAA,qD;MAAA,mB;QAK0C,kB;O;KAL1C,C;+FAOA,yB;MAAA,+D;MAAA,mB;QAMwD,uB;O;KANxD,C;2FAQA,yB;MAAA,+D;MAAA,mB;QAMoD,uB;O;KANpD,C;IAQA,mC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAaXB,GAAyC,iBAAU,sBAaKB,QAAIB,EAAwC,IAAxC,CAAV,C;K;IAE7C,iC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAaXB,GAAyC,iBAAU,sBAaKB,QAAIB,EAAwC,IAAxC,CAAV,C;K;IAE7C,gC;MAI2D,OAAI,eAAJ,GAAqB,OAAO,OAAP,CAArB,GAA0C,W;K;IAErG,mC;MAImE,OAAS,cAAT,QAAS,C;K;gFAE5E,yB;MAAa,gE;MAbA,6B;QAYBI,WAAW,eAduE,IAcvE,C;QWCX,iBAAc,CAAd,UXfkF,IWelf,U;UXA6B,eAf2D,IAevD,CWCtB,KXDsb,CAAJ,C;;QAFyC,OAGB/D,I;O;KA3BX,C;8FAaA,yB;MAAA,gE;MAAA,6B;QAYI,WAAW,eAAa,IAAb,C;QWCX,iBAAc,CAAd,UXAO,IWAP,U;UXA6B,eAAI,KWCtB,KXDsb,CAAJ,C;;QAC7B,OAAO,I;O;KAdX,C;wFAiBA,yB;Me1FA,+D;Mf0FA,gC;QetF0B,gBAAf,gB;QfsGkB,aW3FzB,W;QX2FA,OW1FO,SIZoC,Q;O;KfsF/C,C;yFAwBA,yB;Me3GA,4E;MAAA,gE;Mf2GA,0C;QevGI,qBf2HyB,Qe3HzB,C;QAC8B,gBAAvB,ef0HkB,Qe1HIB,C;Qf0H4B,aWvHnC,W;QXuHA,OWtHO,SIJ4C,Q;O;KfsGvD,C;IAiCl,mC;MAAQ,uBAAG,iBAAO,CAAP,IAAH,C;K;IAQR,qC;MAAQ,OAAA,SAAK,KAAL,GAAY,CAAZ,I;K;4FAEZ,qB;MAK4D,QAAC,mB;K;kGAE7D,qB;MAWI,OAAO,qBAAgB,SAAK,U;K;sFAGhC,yB;MAAA,qD;MAAA,4B;QAKgE,uCAAQ,W;O;KALxE,C;sFAOA,yB;MAAA,qD;MAAA,4B;QAKoD,uCAAQ,W;O;KAL5D,C;sFAOA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,GAAmC,S;K;4FAGvC,+B;MAQoH,OAAA,SAAK,qBAAY,QAAZ,C;K;IAGzH,uC;MAK+E,kBAAhB,0B;MAAwB,+B;MAAxB,OW5MpD,W;K;IX+MX,yC;MAAkD,QAAM,cAAN,C;aAC9C,C;UAD8C,OACzC,W;aACL,C;UAF8C,OAEzC,OAAO,sBAAK,CAAL,CAAP,C;;UAFyC,OAGtC,S;;K;IAGZ,8D;MAGbKe,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACjG,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GA AJ,C;QACb,UAAU,cAAc,MAAd,EAAsb,OAAtB,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,4E;MAe8E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC7G,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GA AJ,C;QACb,UAAU,UAAW,SAAQ,MAAR,EAAgB,OAAhB,C;QAErB,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,8C;MAMQ,gBAAY,OA AZ,C;QAAuB,MAAM,gCAAYB,gBAaA,SAAb,mCAAKD,OAAID,OAAzB,C;WAC7B,gBAAY,CAAZ,C;QAAiB,MAAM,8BAA0B,gBAaA,SAAb,yBAA1B,C;WACvB,cAAU,IAAV,C;QAAkB,MAAM,8BAA0B,cAAW,OAAAX,gCAA2C,IA



A3C,OAA1B,C;K;IAchC,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;IAE1C,8B;MAEoC,MAAM,wBAAoB,8B  
AApB,C;K;;;wF2GjB1C,yB;M1GgCA,wE;M0GhCA,uC;QAmBW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;  
QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UACIB,  
W1GuKJ,a0GvKgB,G1GuKhB,E0GrMyC,SA8BIB,CAAU,GAAV,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uBAAu  
B,CAAC,WAAy,mBAAY,GAAZ,CAAnE,C1GuKvB,C;;Q0GrMA,OAgCO,W;O;KAnDX,C;4FAsBA,6C;MAwBc  
,Q;MAAA,OAAA,SAAK,iB;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAAk  
B,sBAAY,GAAZ,C;QACIB,W1GuKJ,a0GvKgB,G1GuKhB,E0GvKuB,UAAU,GAAV,EAAe,WAAf,EAA4B,CAA  
5B,EAA+B,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAAnE,C1GuKvB,C;;M0GrKA,OAAO,W;K;iFAGX,yB;M  
AAA,gB;MAAA,8B;M1GtBA,wE;M0GsBA,6D;QAnCW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAAf,O  
AAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UA8BwE,U;UA7  
B1F,W1GuKJ,a0GvKgB,G1GuKhB,E0G1IkC,UA7BD,GA6BC,EA7BoB,uBAAuB,CAAC,WAAy,mBAAY,GAA  
Z,CA6BzC,GAAW,qBA7B3B,GA6B2B,EA7BT,CA6BS,CAAX,GAA6C,UA7BxD,WA6BwD,6DAA5D,EA7BiB,  
CA6BjB,C1G0IIC,C;;Q0G3IA,OA1BO,W;O;KAGX,C;kFA0BA,yB;MAAA,gB;MAAA,8B;MAAA,0E;QAICc,Q;  
QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6DQ,WA7D  
U,WAAy,GAAZ,C;UA6DuF,U;UAAjG,W1G2GZ,a0GvKgB,G1GuKhB,E0G3GiD,UA5DhB,GA4DgB,EA5DK,u  
BAAuB,CA4DjE,WA5D8E,mBAAY,GAAZ,CA4D1B,GAAW,qBA5D1C,GA4D0C,EA5DxB,CA4DwB,CAAX,G  
AA6C,UA5DvE,WA4DuE,6DAA5D,EA5DE,CA4DF,C1G2GjD,C;;Q0G5GA,OACY,W;O;KA7BhB,C;iFAgCA,y  
B;MAAA,gB;MAAA,8B;M1GhFA,wE;M0GgFA,qD;QA7FW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QA  
Af,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UAKFiD,U;  
UAjFnE,W1GuKJ,a0GvKgB,G1GuKhB,E0GtFgC,UajFsB,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAiFhD,kB  
AA6B,UajFjC,WaiFiC,6DAAvC,EajFmB,CAiFnB,C1GsFhC,C;;Q0GvFA,OA9EO,W;O;KA6DX,C;oFAoBA,yB;  
MAAA,gB;MAAA,8B;MAAA,kE;QAtFc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,  
sBAAM,CAAN,C;UACV,kBA2GQ,WA3GU,WAAy,GAAZ,C;UA2GgE,U;UAA1E,W1G6DZ,a0GvKgB,G1GuKh  
B,E0G7D+C,UA1GO,uBAAuB,CA0GjE,WA1G8E,mBAAY,GAAZ,CA0GjC,kBAA6B,UA1GhD,WA0GgD,6DA  
AvC,EA1GI,CA0GJ,C1G6D/C,C;;Q0G9DA,OACY,W;O;KAvBhB,C;qFA0BA,yB;MAAA,gB;MAAA,8B;M1G9H  
A,wE;M0G8HA,uC;QA3IW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;U  
ACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UACC,oB;UAKIc,U;UAAjC,IAIIkD,uBAAuB,  
CAAC,WAAy,mBAAY,GAAZ,CAkItF,C;YADA,mBAjI+C,C;;YaiI/C,mBACKB,UAIIW,GAKIX,EAAe,UAIIC,  
WAKID,6DAAf,EAI6B,CAkI7B,C;;UAIIB,W1GuKJ,a0GvKgB,G1GuKhB,mB;;Q0GvCA,OA9HO,W;O;KA2GX,  
C;sFAwBA,yB;MAAA,gB;MAAA,8B;MAAA,oD;QAxIc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB  
;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6JQ,WA7JU,WAAy,GAAZ,C;UACC,oB;UA8Jc,U;UAAjC,IA9Jk  
D,uBAAuB,CA4JjE,WA5J8E,mBAAY,GAAZ,CA8JtF,C;YADA,mBA7J+C,C;;YA6J/C,mBACKB,UA9JW,GA8JX  
,EAAe,UA9JC,WA8JD,6DAAf,EA9J6B,CA8J7B,C;;UAFV,W1GWZ,a0GvKgB,G1GuKhB,mB;;Q0GXA,OAAy,  
W;O;KAvBhB,C;IA6BA,6C;MArKc,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sB  
AAM,CAAN,C;QACV,kBA+KG,WA/Ke,WAAy,GAAZ,C;QA2GgE,U;QAoE/E,W1GPP,a0GvKgB,G1GuKhB,E0  
GomC,CA9KmB,uBAAuB,CA8KtE,WA9KmF,mBAAY,GAAZ,CA0GjC,GAoErC,CAPeQc,GAA6B,UA1GhD,W  
A0GgD,6DAoEnD,IAAM,CAAN,I1GPnC,C;;M0GOA,OAAO,W;K;IgeNp0B,oC;MAAC,kB;MAAuB,kB;K;;wCA  
N7D,Y;MAMsC,iB;K;wCANtC,Y;MAM6D,iB;K;0CAN7D,wB;MAAA,wBAMsC,qCANtC,EAM6D,qCAN7D,C;  
K;sCAAAY;MAAA,OAMsC,mDANtC,IAM6D,wCAN7D,O;K;sCAAAY;MAAA,c;MAMsC,sD;MAAuB,sD;MA  
N7D,a;K;oCAAAY,iB;MAAA,4IAMsC,sCANtC,IAM6D,sCAN7D,I;K;wFrKEA,yB;MAAA,kC;MAAA,4C;MAAA,  
kD;QAMuF,wC;O;MANvF,4CAOI,Y;QAAuC,8B;O;MAP3C,8E;MAAA,2B;QAMuF,2C;O;KANvF,C;IAcsC,2C;  
MAAC,wC;K;0CACnC,Y;MAAqD,4BAAiB,wBAAjB,C;K;;IAIzD,yC;MAI4D,OAAI,oCAAJ,GAA2B,SAAK,KA  
AhC,GAA0C,I;K;IAEtG,uD;MAI0E,OAAI,oCAAJ,GAA2B,SAAK,KAAhC,GAA0C,S;K;IAGpH,8B;MAMoB,Q;  
MADhB,aAAa,gB;MACG,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACL,OAAP,MAAO,EAAO,OAAP,C;;M  
AEX,OAAO,M;K;IAGX,4B;MAUiB,Q;MAHb,mBAAmB,mCAAwB,EAAxB,C;MACnB,YAAy,iBAAa,YAAb,C;  
MACZ,YAAy,iBAAa,YAAb,C;MACC,2B;MAAb,OAAa,cAAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,  
C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;wFUxDX,qB;MAKqE,gB;K;IAErE,i  
C;MAMoE,4BAAiB,SAAjB,C;K;uFAEpE,gC;MAKI,OAAGB,mBAAhB,C;QAAGB,8B;QAAM,UAAU,OAAY,C;;

K;IAMY,oC;MAAC,0B;MACnC,eAAoB,C;K;yCACpB,Y;MAAwC,OAAA,eAAS,U;K;sCACjD,Y;MAA6E,Q;MAAhC,wBAAa,oBAAmB,mBAAAnB,EAAMb,2BAAAnB,QAAAb,EAA0C,eAAS,OAAAnD,C;K;;sF4J5BjD,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAUiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAVjC,C;wFAY A,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAWiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAXjC,C;sFAaA,+C;MAQI,SAAK,aAAI,QAAS,aAAAb,EAAMb,KAAAnB,C;K;ICnCT,8C;MAUI,IAAI,wCAAJ,C;QACI,OAAO,SAAK,4BAAqB,GAARb,C;MAET,4B;M5KoTI,Q;MALX,YAAY,oB4K/Sa,G5K+Sb,C;MACZ,IAAI,iBAAiB,CAAC,4B4KhTG,G5KgTH,CAAtB,C;Q4KhTgC,MAAM,2BAAuB,wCAAvB,C;;Q5KoTIC,2BAAO,sE;;M4KpTX,+B;K;IAGJ,8C;MAUQ,kBADE,SACF,kB;QADJ,OACkC,YAAT,SAAK,IAAI,EAAY,YAAZ,C;;QADIC ,OAEY,uBAAmB,SAAnB,EAAYB,YAAzB,C;K;IAGhB,gD;MAWQ,kBADE,SACF,yB;QADJ,OACyC,cAAT,SA AK,IAAI,EAAY,YAAZ,C;;QADzC,OAEY,8BAA0B,SAA1B,EAAGC,YAAhC,C;K;,,,,;IAc0B,4C;MAAC,wB;M AAoC,0B;K;qEAApC,Y;MAAA,yB;K;0CACvC,iB;MAA4C,OAAI,OAAJ,QAAL,EAAO,KAAP,C;K;4CAChD,Y; MAA+B,OAAI,SAAJ,QAAL,C;K;4CACnC,Y;MAAkC,OAAA,QAAL,W;K;0FACf,Y;MAAQ,OAAA,QAAL,K;K;2 CACnC,Y;MAAkC,OAAA,QAAL,U;K;qDACtC,e;MAA4C,OAAA,QAAL,mBAAAY,GAAZ,C;K;uDACHD,iB;MAA gE,OAAA,QAAL,qBAAc,KAAd,C;K;6CACpE,e;MAA+B,OAAA,QAAL,WAAI,GAAJ,C;K;0FACT,Y;MAAQ,OA AA,QAAL,K;K;4FACH,Y;MAAQ,OAAA,QAAL,O;K;6FACJ,Y;MAAQ,OAAA,QAAL,Q;K;8DAEvD,e;MAAmD,g BAAJ,Q;MAAI,4B;M5K4PxC,Q;MALX,YAAY,oB4KvPyD,G5KuPzD,C;MACZ,IAAI,iBAAiB,CAAC,4B4KxP+ C,G5KwP/C,CAAtB,C;QACI,2B4KzPwE,mB;;Q5K4PxE,2BAAO,sE;;M4K5PoC,+B;K;;IAGN,mD;MAAC,wB;M AA2C,0B;K;4EAA3C,Y;MAAA,yB;K;iDAC1C,iB;MAA4C,OAAI,OAAJ,QAAL,EAAO,KAAP,C;K;mDACHD,Y; MAA+B,OAAI,SAAJ,QAAL,C;K;mDACnC,Y;MAAkC,OAAA,QAAL,W;K;iGACf,Y;MAAQ,OAAA,QAAL,K;K;k DACnC,Y;MAAkC,OAAA,QAAL,U;K;4DACtC,e;MAA4C,OAAA,QAAL,mBAAAY,GAAZ,C;K;8DACHD,iB;MA AgE,OAAA,QAAL,qBAAc,KAAd,C;K;oDACpE,e;MAA+B,OAAA,QAAL,WAAI,GAAJ,C;K;iGACF,Y;MAAQ,O AAA,QAAL,K;K;mGACH,Y;MAAQ,OAAA,QAAL,O;K;oGACU,Y;MAAQ,OAAA,QAAL,Q;K;sDAE5E,sB;MAA yC,OAAA,QAAL,aAAI,GAAJ,EAAS,KAAT,C;K;uDAC7C,e;MAAkC,OAAA,QAAL,cAAO,GAAP,C;K;yDACtC, gB;MAA2C,QAAL,gBAAO,IAAP,C;K;gDAC/C,Y;MAAuB,QAAL,Q;K;qEAE3B,e;MAAmD,gBAAJ,Q;MAAI,4B; M5KuOxC,Q;MALX,YAAY,oB4KIOyD,G5KkOzD,C;MACZ,IAAI,iBAAiB,CAAC,4B4KnO+C,G5KmO/C,CAAt B,C;QACI,2B4KpOwE,mB;;Q5KuOxE,2BAAO,sE;;M4KvOoC,+B;K;;I5KvFnD,oB;MAAA,wB;MACI,8C;K;gCA EA,iB;MAA4C,oCAAsB,KAAM,U;K;kCACxE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K ;iCAC/B,Y;MAAkC,W;K;2CAEIC,e;MAA+C,Y;K;6CAC/C,iB;MAAsD,Y;K;mCACtD,e;MAAwC,W;K;mFACY, Y;MAAQ,6B;K;gFAC/B,Y;MAAQ,6B;K;kFACI,Y;MAAQ,8B;K;uCAEjD,Y;MAAiC,6B;K;;IAjBrC,gC;MAAA,+ B;QAAA,c;;MAAA,wB;K;IAoBA,oB;MAMuE,Q;MAA7B,OAA6B,uE;K;IAEvE,wB;MAaI,OAAI,KAAM,OAA N ,GAAa,CAAjB,GAA0B,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAIB,CAAd,CAAN,CAA1B,GAA6E,U; K;kFAEjF,yB;MAAA,oD;MAAA,mB;QA08C,iB;O;KAP9C,C;8FASA,yB;MAAA,wE;MAAA,mB;QAQ4D,2B;O; KAR5D,C;IAUA,+B;MAYiD,gBAA7C,qBAAoB,YAAY,KAAM,OAAIB,CAApB,C;MAAqD,wB;MAArD,OUJO, S;K;wFVMX,yB;MAAA,4D;MAAA,mB;QA0sD,qB;O;KAPtD,C;IASA,4B;MAM8G,gBAAvC,eAAc,YAAY,KA AM,OAAIB,CAAd,C;MAA+C,wB;MAA/C,OuRb5D,S;K;4FVuBX,yB;MAAA,wE;MAAA,mB;QAK8D,2B;O;KA L9D,C;IAOA,8B;MAU+E,OAAM,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAIB,CAAd,CAAN,C;K;sFA ErF,yB;MchBA,wE;MdgBA,gC;QcZiC,gBAAtB,oB;Qd8BiB,aU7DxB,W;QV6DA,OU5DO,SI8B2C,Q;O;KdYtD,C ;uFA0BA,yB;McnCA,uE;MdmCA,0C;Qc/ByC,gBAA9B,mBdqDiB,QcrDjB,C;QdqD2B,aU3FIC,W;QV2FA,OU1F O,SIqCmD,Q;O;Kd+B9D,C;4FAoCA,qB;MAK+D,QAAC,mB;K;kGAHEh,qB;MAWI,OAAO,qBAAgB,mB;K;sFA G3B,yB;MAAA,oD;MAAA,4B;QAM2D,uCAAQ,U;O;KANnE,C;sFAQA,mC;MASI,OAAI,mBAAJ,GA Ae,cAAf, GAAMC,S;K;yFAEvC,yB;MAyBA,kC;MAAA,8B;MAzBA,iC;QAgCiC,Q;QAx2E,OAwBxD,CAAnB,wDAAMb ,oBAxBoE,GAwBpE,C;O;KAhCpD,C;+EAUA,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAKiC,Q;QAA7B,OAAgD, CAAnB,wDAAMb,YAAI,GAAJ,C;O;KALpD,C;+EAOA,iC;MAKI,sBAAI,GAAJ,EAAS,KAAT,C;K;4FAGJ,yB; MAAA,kC;MAAA,8B;MAAA,iC;QA0iC,Q;QAA7B,OAAgD,CAAnB,wDAAMb,oBAAAY,GAAZ,C;O;KAPpD,C; gGASA,4B;MASsG,OAAA,SAAK,qBAAc,KAAd,C;K;kFAG3G,yB;MAAA,gD;MAAA,8B;MAAA,iC;QASiC,Q; QAA7B,OAAuD,CAA1B,+DAA0B,eAAO,GAAP,C;O;KAT3D,C;6FAWA,qB;MAWoE,oB;K;6FAEpE,qB;MAW oE,sB;K;kFAEpE,yB;MAAA,6B;MAAA,4B;QAIGe,qBAAK,aAAL,EAAU,eAAV,C;O;KAJhE,C;2FAMA,wC;MA MiF,Q;MAAA,mCAAI,GAAJ,oBAAAY,c;K;uGAG7F,yB;MAAA,gB;MAAA,8B;MAAA,+C;QAMe,Q;QALX,YA

AY,oBAAI,GAAJ,C;QACZ,IAAI,iBAAiB,CAAC,4BAAY,GAAZ,CAAtB,C;UACI,OAAO,c;;UAGP,OAAO,sE;;O  
;KANf,C;IAUA,oC;MAUkD,uCAAqB,GAARb,C;K;sFAEID,wC;MAUW,Q;MADP,YAAY,oBAAI,GAAJ,C;MAC  
L,IAAI,aAAJ,C;QACH,aAAa,c;QACb,sBAAI,GAAJ,EAAS,MAAT,C;QACA,a;;QAEA,Y;;MALJ,W;K;wFASJ,qB  
;MAMwF,OAAA,iBAAQ,W;K;wFAEhG,qB;MAMgH,OAAA,iBAAQ,W;K;4FAExH,6C;Maq1BoB,Q;MAAA,Ob  
h1BT,iBag1BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qbh1Ba,Wai1Bb,aAAGb,Obj1Be,Iai1B/B,Ebj1BsC,Sai  
1BZ,CAAE,OAAf,CAA1B,C;;Mbj1BhB,OAA6B,W;K;wFAGjC,6C;Ma60BoB,Q;MAAA,Obr0BT,iBaq0BS,W;MA  
AhB,OAAgB,cAAhB,C;QAAgB,yB;Qbr0Ba,Was0Bb,abt0B0B,Sas0BtB,CAAY,OAAZ,CAAJ,EAAyC,Obt0BC,M  
as0B1C,C;;Mbt0BhB,OAA6B,W;K;IAGjC,kC;MAIyB,Q;MAArB,wBAAqB,KAArB,gB;QAAqB,aAAA,KAArB,  
M;QAAK,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MA  
ArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;I  
AIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,  
sBAAI,GAAJ,EAAS,KAAT,C;;K;wFAIR,yB;MAAA,0D;MAAA,uE;MAAA,uC;QASW,kBAAY,mBAAoB,YAA  
Y,cAAZ,CAApB,C;Qa8xBH,Q;QAAA,Obh1BT,iBag1BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ubh1Ba,Wai  
1Bb,aAAGb,Obj1Be,Iai1B/B,Eb/xB2C,Sa+xBjB,CAAE,OAAf,CAA1B,C;;Qb/xBhB,OAlD6B,W;O;KAyCjC,C;oF  
AYA,yB;MAAA,0D;MAAA,uE;MAAA,uC;QAYW,kBAAU,mBAAoB,YAAY,cAAZ,CAApB,C;Qa+wBD,Q;QA  
AA,Obr0BT,iBaq0BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ubr0Ba,Was0Bb,abhxBYc,SagxBrc,CAAY,OA  
AZ,CAAJ,EAAyC,Obt0BC,Mas0B1C,C;;QbhxBhB,OAtD6B,W;O;KA0CjC,C;0FAeA,yB;MAAA,wE;MAAA,uC;  
QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3FsE,QAAQ,W;QA2F5F,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,U  
AAU,KAAM,IAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;  
KAbX,C;8FAGBA,yB;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3GsE,QAAQ,W;QA2  
G5F,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,KAAM,MAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,  
EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;KAbX,C;yFAiBA,6C;MAOoB,Q;MAAA,OAAA,SA3HoE,QAAQ,  
W;MA2H5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,WAAY,aAAI,OAAQ,IA  
AZ,EAAiB,OAAQ,MAAzB,C;;MAGpB,OAAO,W;K;qFAGX,yB;MAAA,wE;MAAA,uC;QAOW,kBAAS,oB;QA  
fa,Q;QAAA,OA3HoE,iBAAQ,W;QA2H5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAcmC,SAd/B,CAAU,OAAV,  
CAAJ,C;YACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX,C;-FAUA,6  
C;MAOoB,Q;MAAA,OAAA,SapJoE,QAAQ,W;MAoJ5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,CAAC,UA  
AU,OAAV,CAAL,C;UACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;MAGpB,OAAO,W;K;2FAG  
X,yB;MAAA,wE;MAAA,uC;QAOW,kBAAY,oB;QafH,Q;QAAA,OapJoE,iBAAQ,W;QAoJ5F,OAAgB,cAAhB,  
C;UAAgB,yB;UACZ,IAAI,CackC,SadjC,CAAU,OAAV,CAAL,C;YACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OA  
AQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX,C;IAUA,0B;MAQqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,QAA  
M,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAakB,sBAAK,CAAL,CAAIB,GAA+  
B,oBAAW,OAAhD,C;YAAL,K;;YACQ,0BAAM,qBAAoB,YAAY,cAAZ,CAApB,CAAN,C;YAHL,K;;QAAP,W;;  
MAMJ,OAAoC,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAGxC,yC;MAIwB,SAApB,WAAoB,Y;MAApB,kB;K;  
IAEJ,4B;MAM6D,QAAM,gBAAN,C;aACzD,C;UADyD,OACpD,U;aACL,C;UAFyD,OAEPD,MAAM,UAAK,CA  
AL,CAAN,C;;UAFoD,OAGjD,mBAAM,qBAAoB,YAAY,gBAAZ,CAApB,CAAN,C;;K;IAGZ,yC;MAIwB,OAAP  
B,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAM4D,OAA6B,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAEzF,yC;MAI  
wB,SAApB,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAMqD,QAAM,cAAN,C;aACjD,C;UADiD,OAC5C,U;aACL,C;  
UAFiD,OC/X8B,uB;;Ud+X9B,OAGzC,uB;;K;IAGZ,iC;MAMmE,4BAAC,SAAd,C;K;IAEnE,yC;MAKI,WAAoB,0  
B;MAApB,kB;K;IAEJ,kC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAAoB,MAAM,IAAN,C;;QAAqC,kB  
AApB,qBAAC,SAAd,C;QAA4B,wBAAS,UAAT,EAAqB,WAArB,C;QAAjE,OUhiBO,W;;MVgiBP,W;K;IAEJ,mC  
;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,MAAN,KAAM,C;;QAAiC,kBAAPB,qBAAC,SAAd,C;Q  
AA4B,4B;QAAnE,OUziBO,W;;MVyiBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,  
QAAN,KAAM,C;;QAAiC,kBAAPB,qBAAC,SAAd,C;QAA4B,0B;QAAnE,OUljBO,W;;MVkjBP,W;K;IAEJ,mC;M  
AOwB,kBAAPB,qBAAC,SAAd,C;MAA4B,4B;MAA5B,OAA4C,oBU3jBrC,WV2jBqC,C;K;IAEhD,iC;MAOwB,k  
BAAPB,qBAAC,SAAd,C;MAA4B,+B;MAA5B,OUpkBO,W;K;0FVukBX,2B;MAKI,sBAAI,IAAK,MAAT,EAAG  
B,IAAK,OAArB,C;K;4FAGJ,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,  
gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;

O;KALJ,C;4FAQA,0B;MAKI,yBAAO,GAAP,C;K;IAGJ,kC;MAOwB,kBAAf,aAAL,SAAK,C;MA sCL,6B;MA tC A,OAA+C,oBUtnBxC,WVsnBwC,C;K;IAEnD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAqCK,YAAL,gBAAK,O; MA rCV,OAAgD,oBUhoBzC,WVgoByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAoCK,YAAL,gBA AK,O;MApCV,OAAgD,oBU1oBzC,WV0oByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAMCK,YAA L,gBAAK,O;MAnCV,OAAgD,oBUppBzC,WVopByC,C;K;4FAEpD,0B;MAMI,uBAAO,GAAP,C;K;8FAGJ,yB; MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KA AK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,kC;QAMc,UAAV,SA AK,KA AK,EAAU,IAAV,C;O;KANd,C;IAUA,wC;MACsD,QAAM,cAAN,C;aACID,C;UADkD,OAC7C,U;aACL, C;UAFkD,gB;;UAAA,OAG1C,S;;K;oF6KtwBZ,yB;MAAA,8D;MAAA,8B;MAAA,qC;QAUiC,Q;QAA7B,OAA2 D,CAA9B,sEAA8B,eAAO,OAAP,C;O;KAV/D,C;wFAYA,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASiC,Q;QAA7 B,OAA2D,CAA9B,sEAA8B,oBAAU,QA AV,C;O;KAT/D,C;wFAWA,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASi C,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QA AV,C;O;KAT/D,C;4FAWA,8B;MAKI,SAAK,WAAI,OA AJ, C;K;4FAGT,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD; MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,S AAK,EAAO,QAAP,C;O;KALT,C;8FAQA,8B;MAKI,SAAK,cAAO,OAAP,C;K;8FAGT,yB;MAAA,sD;MAAA,sC ;QAKS,UAAL,SAAK,EAAU,QA AV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAA U,QA AV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QA AV,C;O;KALT,C;IA QA,qC;MAIU,IAIe,I;MAHjB,kBADE,QACF,c;QAAiB,OAAO,yBAAO,QAAP,C;;QAEpB,aAAsB,K;QACT,0B;Q AA b,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,oBAAI,IAAJ,CAAJ,C;YAAe,SAAS,I;;QAC5B,OAAO,M;;K;IAKnB, uC;MAKiB,Q;MADb,aAAsB,K;MACT,0B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,oBAAI,IAAJ,CAAJ,C; UAAe,SAAS,I;;MAE5B,OAAO,M;K;IAGX,uC;MAII,OAAO,yBAAGB,OAAT,QAAS,CAAhB,C;K;IAGX,0C;MA IW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;MAIHG,Q;MAkH7B,OAIH2D,CAA9B,sEAA8B,oBAAU,UAAV, C;K;IAqH/D,0C;MAII,UAAmB,8BAAT,QAAS,C;MACnB,09K0EwD,C8K1EjD,G9K0EkD,U8K1EID,IAAoB,4B AAU,GA AV,C;K;IAG/B,0C;MAII,OpLqoPO,EoLroPA,QpL6jPA,YAAQ,CAwER,CoLroPA,IAAyB,4BAAmB,8B AAT,QAAS,CAAnB,C;K;IAGpC,0C;MAIW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;MA7HG,Q;MA6H7B,O A7H2D,CAA9B,sEAA8B,oBAAU,UAAV,C;K;IAGl/D,0C;MAII,IpLunPO,EoLvnPH,QpL+iPG,CAwER,C oLvnPP,C;QACI,OAAO,4BAAmB,8BAAT,QAAS,CAAnB,C;;QAEp,OAAO,wB;K;IAGf,0C;MAII,UAAmB,8BA AT,QAAS,C;MACnB,I9K0CwD,C8K1CpD,G9K0CqD,U8K1CzD,C;QACI,OAAO,4BAAU,GA AV,C;;QAEp,OA AO,wB;K;IAGf,kC;MACI,a9KmCwD,CAAC,mB;M8K1CzD,iB;MACA,OAAO,M;K;IAIX,2C;MAKkF,gCAAc,S AAd,EAAyB,IAAzB,C;K;IAEIF,2C;MAKkF,gCAAc,SAAd,EAAyB,KAzB,C;K;IAEIF,sE;MACI,iBAAa,KAAb, C;MnKlJgB,kBmKmjX,oB;MACD,OAAO,qBAAP,C;QACI,IAAI,UAAU,kBAAV,6BAAJ,C;UACI,oB;UACA,W AAS,I;;MAGrB,OAAO,Q;K;oFAIX,4B;MAM6D,kCAAS,KAAT,C;K;IAE7D,gC;MAKiD,IAAI,mBAAJ,C;QAAe, MAAM,2BA AuB,gBAAvB,C;;QAARb,OAAMe,2BAAS,CAAT,C;K;IAEpH,sC;MAKwD,OAAI,mBAAJ,GA Ae,I AAF,GAAYB,2BAAS,CAAT,C;K;IAEjF,+B;MAKgd,IAAI,mBAAJ,C;QAAe,MAAM,2BA AuB,gBAAvB,C;;QAA rB,OAAMe,2BAAS,2BAAT,C;K;IAEnH,qC;MAKuD,OAAI,mBAAJ,GA Ae,IAAf,GAAYB,2BAAS,2BAAT,C;K;I AEhF,2C;MAK8E,kCAAc,SAAd,EAAyB,IAAzB,C;K;IAE9E,2C;MAK8E,kCAAc,SAAd,EAAyB,KAzB,C;K;IA E9E,wE;MAEgB,UAGS,MAHT,EAcY,MA dZ,EAc6B,M;MAfzC,IAAI,uCAAJ,C;QACI,OAAoC,cAA5B,sEAA4B, EAAC,SAAd,EAAyB,uBAzB,C;MAExC,iBAAsB,C;MACD,oC;MAArB,qBAakB,CAAIB,mC;QACI,cAAc,sBA AK,SAAL,C;QACd,IAAI,UAAU,OAAV,MAAsB,uBAA1B,C;UACI,Q;QAEJ,IAAI,eAAc,SAIIB,C;UACI,sBAA K,UAAAL,EAAmB,OAAnB,C;QAEJ,+B;;MAEJ,IAAI,aAAa,cAAjB,C;QACwB,oC;QAAiB,mB;QAARc,oE;UACI, 2BAAS,WAAT,C;QAEJ,OAAO,I;;QAEp,OAAO,K;;K;IChS+B,wC;MAAkC,uB;MAAjC,0B;K;4FACpB,Y;MAA Q,OAAA,eAAS,K;K;iDACxC,iB;MAAkC,mCAAS,0BAAoB,KAAPB,CAAT,C;K;;IAGT,gC;MAAyC,8B;MAAx C,0B;K;oFACH,Y;MAAQ,OAAA,eAAS,K;K;yCACxC,iB;MAAkC,mCAAS,0BAAoB,KAAPB,CAAT,C;K;mCA EIC,Y;MAAuB,eAAS,Q;K;8CAChC,iB;MAAuC,OAAA,eAAS,kBAAS,0BAAoB,KAAPB,CAAT,C;K;yCAEHd,0 B;MAA8C,OAAA,eAAS,aAAI,0BAAoB,KAAPB,CAAJ,EAAgC,OA AhC,C;K;yCACvD,0B;MACI,eAAS,aAAI,2 BAAqB,KAARb,CAAJ,EAAiC,OAAjC,C;K;;IAIjB,+C;MACoB,Q;MAAA,kC;MAAhB,IAAa,CAAT,0BAAJ,C;Q AAA,OAA2B,8BAAY,KA AZ,I;;QAAuB,MAAM,8BAA0B,mBAAGB,KA AhB,2BAA0C,gBAAG,2BAAH,CAA1 C,OAA1B,C;K;IAE5D,gD;MACoB,Q;MAAA,qB;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAAsB,iBAAO,KAAP

,I;;QAAkB,MAAM,8BAA0B,oBAAiB,KAAjB,2BAA2C,gBAAG,cAAH,CAA3C,OAA1B,C;K;IAGID,+B;MAK+  
C,gCAAqB,SAArB,C;K;IAE/C,iC;MAM6D,wBAAa,SAAb,C;K;;;IxKpC7D,oD;MAQuF,wC;K;IARvF,8CASI,Y;  
MAAuC,8B;K;IAT3C,gF;IyKY8G,wC;MAAA,mB;QAAE,kBAAS,aAAT,C;O;K;IAThH,yB;MASqG,oCAAS,sBA  
AT,C;K;8FAErG,yB;MAAA,kD;MzKdA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;  
O;MAT3C,8E;MyKiB2I,qD;QAAA,mB;UAAE,gBAAS,qBAAT,C;S;O;MAH7I,gC;QAGkI,kCAAS,mCAAT,C;O;  
KAHII,C;IAKA,2B;MAQI,eAAe,6B;MACf,oBAA0B,+BAAN,KAAM,EAAwC,QAAXC,EAA+D,QAA/D,C;MAC  
1B,OAAO,Q;K;8FAGX,yB;MAAA,kD;MAAA,gC;QAGkI,gBAAS,aAAT,C;O;KAHII,C;IAGB0C,yB;K;+CAoBtC,  
kC;MAOI,IAAI,uCAA0B,QAAS,UAAvC,C;QAAkD,M;MACID,OAAO,sBAAS,QAAS,WAAIB,e;K;+CAGX,kC;  
MAQqD,6BAAS,QAAS,WAAIB,e;K;;;;IAezD,mC;MAA2C,wB;MACvC,eAAoB,C;MACpB,mBAA4B,I;MAC  
5B,sBAAYC,I;MACzC,gBAAoC,I;K;gDAEpC,Y;MACI,OAAO,IAAP,C;QACI,QAAM,YAAN,C;eACI,C;YAAA,  
K;eACA,C;YACI,IAAI,kCAAe,UAAAnB,C;cACI,eAAQ,C;cACR,OAAO,I;;cAEP,sBAAe,I;;YALvB,K;eAOA,C;Y  
AAc,OAAO,K;eACrB,C;eAAA,C;YAAgC,OAAO,I;;YAC/B,MAAM,yB;;QAGIB,eAAQ,C;QACR,WAAW,4B;Q  
ACX,gBAAW,I;QACX,IzH/FR,oBDgDQ,W0H+CY,kB1H/CZ,CChDR,C;;K;6CyHmGA,Y;MACU,IASe,I;MATrB  
,QAAM,YAAN,C;aACI,C;aAAA,C;UAAsC,OAAO,qB;aAC7C,C;UACI,eAAQ,C;UACR,OAAO,kCAAe,O;aAE1  
B,C;UACI,eAAQ,C;UACR,aACa,mF;UACb,mBAAY,I;UACZ,OAAO,M;;UAEH,MAAM,yB;;K;uDAlTb,Y;MACI  
,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;;QAA8B,OAAO,W;K;2DAG/D,Y;MAA4C,QAAM,YAAN,C;aACxC,  
C;UADwC,OAC1B,6B;aACd,C;UAFwC,OAExB,6BAAsB,sBAATb,C;;UAFwB,OAGhC,6BAAsB,uCAAoC,YAA  
1D,C;;K;IAOqC,4E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;oDALR,+B;MACI,mBAAY,K;MACZ  
,eAAQ,C;MACR,OAA6C,0CAAtC,c;K;IAUsC,+E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;yDAN  
R,kC;MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,M;MACzB,sBAAe,Q;MACf,eAAQ,C;MACR,OAA6C,6CAAt  
C,c;K;2DAMX,kB;M1HjBO,Q;MADP,e0HoBI,M1HpBJ,C;MACO,Q0HmBH,M1HnBG,+D;M0HoBH,eAAQ,C;K  
;kGAIR,Y;MAAQ,0C;K;;IzK1LhB,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;sFAAA,yB;MA  
AA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;O;MAT3C,8E;MAAA,2B;QAQuF,2  
C;O;KARvF,C;IAiBGe,+C;MAAA,mB;QAAE,sB;O;K;IALIE,kC;MAKuD,OAAkB,2CAAT,+BAAS,E;K;IAEzE,8  
B;MAK6D,OAAI,Qb2rPtD,YAAQ,Ca3rP0C,GAAwB,eAAxB,GAAsD,WAAT,QAAS,C;K;IAEnH,yB;MAG8C,kC  
;K;IAE9C,yB;MAAA,6B;K;uCACI,Y;MAA6C,kC;K;2CAC7C,a;MAA4B,kC;K;2CAC5B,a;MAA4B,kC;K;;IAHh  
C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;oFAMA,yB;MAAA,2D;MAAA,4B;QAM4D,uCAAQ,e;O;KANpE,C;I  
AgB4F,mH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wD;MAAA,kC;K;;;kDAAA,Y;;;cACxFeA  
Ae,uBAAa,W;cAC5B,IAAI,QAAS,UAAb,C;gBACI,gB;gCAAA,sCAAS,QAAT,O;oBAAA,2C;yBAAA,yB;gBAA  
A,Q;;gBAEA,gB;gCAAA,sCAAS,iCAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;;cAJJ,W;;cAAA,W;;;  
K;IADwF,gE;MAAA,yD;uBAAA,uG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAP5F,4C;MAOmF,gBAAS,uCAAT,  
C;K;IAGbB,4B;MAAE,OAAA,EAAG,W;K;IAP3E,8B;MAO8D,4BAAQ,cAAR,C;K;IAUQ,8B;MAAE,OAAA,EA  
AG,W;K;IAR3E,8B;MAQ8D,4BAAQ,gBAAR,C;K;IAM1B,8B;MAAE,S;K;IAJtC,wC;MAEgB,Q;MADZ,IAAI,8  
CAAJ,C;QACI,OAA4C,CAApC,2EAAoC,kBAAQ,QAAR,C;;MAEHd,OAAO,uBAAMb,SAAnB,EAAYB,gBAAz  
B,EAaiC,QAAjC,C;K;IAGX,4B;MAYiB,Q;MAFb,YAAY,gB;MACZ,YAAY,gB;MACC,2B;MAAb,OAAa,cAAb,  
C;QAAA,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAA  
S,KAAT,C;K;IAGX,+B;MAQqD,6BAAS,4BAAT,C;K;IAW0B,+G;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,0C;  
MAAA,4C;MAAA,0B;MAAA,kC;K;;;mDAAA,Y;;;kCAC9D,0C;cACb,gB;;;cAAA,IAAO,iBPyFkD,UOzFzD,  
C;gBAAA,gB;;;cACI,QAAQ,yBAAO,iBAAQ,iBAAO,KAAf,C;cACf,WAAkB,WAAP,iBAAO,C;cACIB,YAAgB,  
IAAI,iBAAO,KAAf,GAAqB,iBAAO,aAAI,CAAJ,EAAO,IAAP,CAA5B,GAA8C,I;cAC1D,gB;8BAAA,iCAAM,K  
AAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAJJ,gB;;cAMJ,W;;;K;IAR+E,4D;MAAA,yD;uBAAA,mG;Y  
AAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAT/E,uC;MASmE,gBAAY,kCAAZ,C;K;IAkBhC,0D;MAE/B,wB;QAAA,WA  
AgC,I;MADhC,0B;MACA,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,  
E;MACrB,gBAAMb,I;K;oEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCA  
AU,IAAV,MAAMb,sCAAvB,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ,M;;MAGR,iBAAY,C;K;8DAGhB,Y;  
MASW,Q;MARP,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aA  
Aa,a;MACb,gBAAW,I;MACX,iBAAY,E;MAEZ,OAAO,yE;K;IEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB  
;MACJ,OAAO,mBAAa,C;K;;2CAhC5B,Y;MAAuC,yD;K;;IA2C3C,qD;MAAY,0B;MAAMC,gC;K;IACJ,gF;MAA

A,0D;MACnC,gBA Ae,oCAAS,W;K;iEACxB,Y;MACI,OAAO,6CAAY,aAAS,OAArB,C;K;oEAGX,Y;MACI,OA  
AO,aAAS,U;K;;8CAPxB,Y;MAAuC,4D;K;qDAWvC,oB;MACI,OAAO,uBAA4B,eAA5B,EAAsC,kBAAtC,EAA  
mD,QAAAnD,C;K;;IAUf,4D;MAAY,0B;MAAmC,gC;K;IACJ,8F;MAAA,wE;MACnC,gBA Ae,2CAAS,W;MACxB,  
aAAY,C;K;wEACZ,Y;MAC0C,Q;MAAtC,OAAO,oDAAY,oBAAmB,iBAAnB,EAAMb,yBAAnB,QAAZ,EAAyC,  
aAAS,OAAID,C;K;2EAGX,Y;MACI,OAAO,aAAS,U;K;;qDARxB,Y;MAAuC,mE;K;;IAkB3C,oC;MAAY,0B;K;I  
AC6C,wE;MACjD,gBA Ae,gCAAS,W;MACxB,aAAY,C;K;6DACZ,Y;MAC2C,Q;MAAvC,OAAO,iBAAa,oBAA  
mB,iBAAnB,EAAMb,yBAAnB,QAAb,EAA0C,aAAS,OAAAnD,C;K;gEAGX,Y;MACI,OAAO,aAAS,U;K;;0CARx  
B,Y;MAAqD,wD;K;;IAMBzD,0D;MACI,4B;MACA,4B;MACA,4B;K;IAEuC,sE;MAAA,gD;MACnC,iBAAgB,gC  
AAU,W;MAC1B,iBAAgB,gCAAU,W;K;4DAC1B,Y;MACI,OAAO,sCAAU,cAAU,OAApB,EAA4B,cAAU,OAAt  
C,C;K;+DAGX,Y;MACI,OAAO,cAAU,UAAV,IAAuB,cAAU,U;K;;yCARhD,Y;MAAuC,uD;K;;IAc3C,6D;MACI,  
0B;MACA,gC;MACA,0B;K;IAEuC,4E;MAAA,sD;MACnC,gBA Ae,kCAAS,W;MACxB,oBAAiC,I;K;+DAEjC,Y;  
MACI,IAAI,CAAC,2BAAL,C;QACI,MAAM,6B;MACV,OAAO,gCA Ae,O;K;kEAG1B,Y;MACI,OAAO,2B;K;+E  
AGX,Y;MACQ,Q;MAAJ,IAAI,iEAA2B,KAA/B,C;QACI,oBAAe,I;MAEnB,OAAO,yBAAP,C;QACI,IAAI,CAAC  
,aAAS,UAAAd,C;UACI,OAAO,K;;UAEP,cAAc,aAAS,O;UACvB,uBAAuB,wCAAS,2CAAY,OAAZ,CAAT,C;UA  
CvB,IAAI,gBAAiB,UAArB,C;YACI,oBAAe,gB;YACf,OAAO,I;;;MAInB,OAAO,I;K;;4CA9Bf,Y;MAAuC,0D;K;  
;IAoC9B,6I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,4C;MAAA,kD;MAAA,gD;MAAA,wB;MAAA,yB;MAAA  
,kC;K;;;yDAAA,Y;;;kBAGyC,I;CAFIC,C;cACI,sD;cAAhB,gB;;;cAAA,KAAgB,yBAAhB,C;gBAAA,gB;;;cA  
AgB,oC;cACZ,aAAa,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAAnB,QAAV,EAAuC,OAAvC,C;cACb,gB;8BAA  
A,sCAAS,4BAAS,MAAT,CAAT,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAIJ,W;;;K;IANS,0F;M  
AAA,yD;uBAAA,i;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IADb,wD;MACI,gBAAS,kDAAT,C;K;;;IAoByB,qD;M  
ACzB,0B;MACA,8B;MACA,0B;MC3TA,IAAI,ED+TQ,qBAAc,CC/TtB,CAAJ,C;QACI,cD8T2B,+CAA4C,iB;QC  
7TvE,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EDgUQ,mBAAy,CChUpB,CAAJ,C;QACI,gBD+TyB,6  
CAA0C,e;QC9TnE,MAAM,gCAAyB,SAAQ,WAAjC,C;;MAFV,IAAI,EDiUQ,mBAAy,iBCjUpB,CAAJ,C;QACI,  
gBDgUkC,0DAAuD,eAAvD,WAAmE,iB;QC/TrG,MAAM,gCAAyB,SAAQ,WAAjC,C;;K;sFDkUa,Y;MAAQ,yB  
AAW,iBAAX,I;K;yCAE/B,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GAAqC,gBAAy,eAAZ,EAASB,oBA  
Aa,CAAb,IAAtB,EAAsC,eAAtC,C;K;yCAC9E,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,IAAhB,GAA0B,gBAAy  
,eAAZ,EAASB,iBAAtB,EAaKc,oBAAa,CAAb,IAAI,C;K;IAEzC,8D;MAAA,wC;MAEtB,gBA Ae,2BAAS,W;M  
ACxB,gBA Ae,C;K;0DAEf,Y;MAEI,OAAO,gBA AW,kCAAX,IAAyB,aAAS,UAAzC,C;QACI,aAAS,O;QACT,qC;  
;K;2DAIR,Y;MACI,a;MACA,OAAQ,gBA AW,gCAAZ,IAAyB,aAAS,U;K;wDAG7C,Y;MACI,a;MACA,IAAI,iBA  
AY,gCAAhB,C;QACI,MAAM,6B;MACV,qC;MACA,OAAO,aAAS,O;K;;qCAvBxB,Y;MAA0B,mD;K;;IAgCA,u  
C;MAC1B,0B;MACA,oB;MC3WA,IAAI,ED+WQ,gBAAS,CC/WjB,CAAJ,C;QACI,cD8WsB,yCAAsC,YAAtC,M  
;QC7WtB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;0CDgXV,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GA  
AqC,gBAAy,eAAZ,EAASB,CAAtB,EAyB,YAAzB,C;K;0CAC9E,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,IAA  
hB,GAA0B,iBAAa,eAAb,EAAuB,CAAvB,C;K;IAE5B,gE;MACnC,YAAW,yB;MACX,gBA Ae,4BAAS,W;K;yDA  
ExB,Y;MACI,IAAI,cAAQ,CAAZ,C;QACI,MAAM,6B;MACV,6B;MACA,OAAO,aAAS,O;K;4DAGpB,Y;MACI,  
OAAO,YAAO,CAAP,IAAY,aAAS,U;K;;sCAZpC,Y;MAAuC,oD;K;;IAS3C,gD;MACI,0B;MACA,4B;K;IAEuC,  
0E;MAAA,oD;MACnC,gBA Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,IAAI,aA  
AS,UAAb,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,CAAJ,C;UACI,iBAAy,C;UACZ,gBA AW,I;U  
ACX,M;;;MAGR,iBAAy,C;K;8DAGhB,Y;MAMiB,Q;MALb,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mB  
AAa,CAAjB,C;QACI,MAAM,6B;MACV,aACa,gF;MAGb,gBA AW,I;MACX,iBAAy,E;MACZ,OAAO,M;K;iEA  
GX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAIC5B,Y;MAAuC,yD;K;;IA2Cb,u  
C;MAC1B,0B;MACA,oB;MC5bA,IAAI,ED+bQ,gBAAS,CC/bjB,CAAJ,C;QACI,cD8bsB,yCAAsC,YAAtC,M;QC  
7btB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;0CDgcV,a;MItXO,SJsXmC,eAAQ,CAAR,I;MAAD,OAA4B,KAAK,  
CAAT,GAAY,yBAAZ,GAAuC,iBAAa,eAAb,EAAuB,EAAvB,C;K;0CACxG,a;MIvXO,SJuXmC,eAAQ,CAAR,I;  
MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,gBAAy,eAAZ,EAASB,YAAtB,EAA6B,EAA7B,C;K;IA  
EjE,gE;MACnC,gBA Ae,4BAAS,W;MACxB,YAAW,yB;K;2DAEX,Y;MAEI,OAAO,YAAO,CAAP,IAAY,aAAS,  
UAA5B,C;QACI,aAAS,O;QACT,6B;;K;yDAIR,Y;MACI,a;MACA,OAAO,aAAS,O;K;4DAGpB,Y;MACI,a;MAC  
A,OAAO,aAAS,U;K;;sCAnBxB,Y;MAAuC,oD;K;;IA6B3C,gD;MACI,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;M

ACnC,gBA Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;gEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;  
QACI,WAAW,aAAS,O;QACpB,IAAI,CAAC,wCAAU,IAAV,CAAL,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ  
,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMqB,Q;MALjB,IAAI,mBAAa,EAAjB,C;QACI,a;MAEJ,IAAI,mBAAa,  
CAAjB,C;QACI,aCa,gF;QACb,gBAAW,I;QACX,iBAAY,C;QACZ,OAAO,M;;;MAEX,OAAO,aAAS,O;K;iEAG  
pB,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,a;MACJ,OAAO,mBAAa,CAAb,IAAkB,aAAS,U;K;;2CAIC1C,Y;MA  
AuC,yD;K;;IAuCN,+C;MAAC,sB;MAAiC,gC;K;0CACnE,Y;MAAuC,4BAAiB,aAAO,WAAxB,EAAoC,kBAAPC  
,C;K;;IAGP,+C;MAAuE,2B;MAAtE,sB;MAAiC,gC;MACIE,kBAAuB,c;K;6CAEvB,Y;MACI,OAAO,aAAO,UAA  
d,C;QACI,WAAW,aAAO,O;QACIB,UAAU,mBAAY,IAAZ,C;QAEV,IAAI,eAAS,WAAI,GAAJ,CAAb,C;UACI,  
mBAAQ,IAAR,C;UACA,M;;;MAIR,W;K;;IAKgC,0D;MAAC,wC;MAAuC,kC;K;IACrC,0E;MAAA,oD;MACnC,  
gBAAmB,I;MACnB,iBAAqB,E;K;oEAERB,Y;MACI,gBA Ae,mBAAa,EAAjB,GAAqB,+CAArB,GAA4C,2CAAa,  
4BAAb,C;MACvD,iBAAgB,qBAAJ,GAASB,CAAtB,GAA6B,C;K;8DAG7C,Y;MAMiB,Q;MALb,IAAI,iBAAY,C  
AAhB,C;QACI,iB;MAEJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,8D;MAEb,iBAAY,E;MACZ,  
OAAO,M;K;iEAGX,Y;MACI,IAAI,iBAAY,CAAhB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAx5B,Y;MAAu  
C,yD;K;;IA6B3C,kC;MAWI,OAAW,iDAAJ,GAAW,C,SAAX,C,GAakD,4BAAwB,SAAXB,C;K;IAeIB,uD;MAAA,  
qB;QAAE,6B;O;K;IAX7C,wC;MAWI,OAA2D,cAApD,sBAakB,YAAIB,EAAgC,qCAAhC,CAAoD,C;K;IAqBrC,  
iD;MAAA,mB;QAAE,mB;O;K;IAIB5B,gD;MAeI,OAAI,YAAJ,GACI,2BADJ,GAGI,sBAakB,+BAAIB,EAA4B,  
YAA5B,C;K;IAER,wD;MAcI,6BAAkB,YAAIB,EAAgC,YAAhC,C;K;ILxpBJ,oB;MAAA,wB;MACI,8C;K;gCAE  
A,iB;MAA4C,oCAAmB,KAAM,U;K;kCACrE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K;i  
CAC/B,Y;MAAkC,W;K;wCACIC,mB;MAAmD,Y;K;6CACnD,oB;MAAmE,OAAA,QAAS,U;K;kCAE5E,Y;MAA  
6C,kC;K;uCAE7C,Y;MAAiC,6B;K;;IArC,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IAkBA,oB;MAIoC,6B;K;IAE  
pC,2B;MAMmD,OAAI,QAAS,OAAT,GAAGB,CAApB,GAAGC,MAAT,QAAS,CAAhC,GAA6C,U;K;iFAEHG,yB  
;MAAA,mD;MAAA,mB;QAKwC,iB;O;KALxC,C;6FAOA,yB;MAAA,uE;MAAA,mB;QAQsD,2B;O;KARtD,C;I  
AUA,kC;MAKiE,OAAS,aAAT,QAAS,EAAa,qBAAC,YAAY,QAAS,OAARb,CAAd,CAAb,C;K;uFAE1E,yB;MA  
AA,2D;MAAA,mB;QAGgD,qB;O;KAHhD,C;IAKA,+B;MAC2D,OAAS,aAAT,QAAS,EAAa,eAAQ,YAAY,QA  
AS,OAARb,CAAR,CAAb,C;K;2FAEpE,yB;MAAA,uE;MAAA,mB;QAMwD,2B;O;KANxD,C;IAQA,iC;MAKmE,O  
AAS,aAAT,QAAS,EAAa,qBAAC,YAAY,QAAS,OAARb,CAAd,CAAb,C;K;IAE5E,+B;MAMyD,OAAI,eAAJ,GA  
AqB,MAAM,OAAN,CAArB,GAAYC,U;K;IAEIG,kC;MAQI,OAAGB,gBAAT,QAAS,EAAgB,sBAAhB,C;K;sFAG  
pB,yB;MavBA,uE;MbuBA,gC;QanB8B,gBAAnB,oB;QbqCiB,aS/CxB,W;QT+CA,OS9CO,SISwC,Q;O;KbmBnD,  
C;wFA0BA,yB;Ma1CA,wE;Mb0CA,0C;QatCsC,gBAA3B,mBb4DiB,Qa5DjB,C;Qb4D2B,aS7EIC,W;QT6EA,OS5  
EO,SIgBgD,Q;O;KbsC3D,C;sFA+BA,yB;MAAA,mD;MAAA,4B;QAEkD,uCAAQ,U;O;KAF1D,C;IAIA,wC;MA  
AgD,QAAM,cAAN,C;aAC5C,C;UAD4C,OACvC,U;aACL,C;UAF4C,OAEvC,MAAM,oBAAW,OAAjB,C;;UAFu  
C,OAGpC,S;;K;IKnKZ,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;I0KLA,yC;MzK4BI,IAAI,E  
yK3BI,OAAO,CAAP,IAAY,OAAO,CzK2BvB,CAAJ,C;QACI,cyK3BI,aAAJ,GACI,yEADJ,GAGI,8C;QzKyBJ,M  
AAM,gCAAYB,OAAQ,WAAjC,C;;K;IyKnBM,mI;MAAA,mB;QAAE,wBAAiB,gCAAjB,EAA6B,YAA7B,EAAM  
C,YAAnC,EAAyC,sBAAZC,EAAyD,mBAAZD,C;O;K;IAFtB,gF;MACI,oBAAoB,IAApB,EAA0B,IAA1B,C;MAC  
A,oCAAGB,6EAAhB,C;K;IAKyB,yL;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wC;MAAA,gD;MA  
AA,sD;MAAA,4D;MAAA,wB;MAAA,0B;MAAA,uB;MAAA,0B;MAAA,wB;MAAA,qB;MAAA,4B;MAAA,kC;  
K;;;2DAAA,Y;;;cACrB,4BAAiC,eAAL,uBAAK,EAAa,IAAb,C;+BACvB,0BAAO,uBAAP,I;cACV,IAAI,kBAA  
O,CAAX,C;oCACiB,iBAAa,qBAAb,C;kCACF,C;gBACD,6C;gBAAV,iB;;;sCAaa,gBAAC,qBAAD,C;gBACH,+C;  
gBAAV,gB;;;cAAA,KAAU,2BAAV,C;gBAAA,gB;;;cAAU,kC;cACN,mBAAO,WAAI,GAAJ,C;cACP,IAAI,m  
BAAO,SAAX,C;gBACI,IAAI,mBAAO,KAAP,GAAC,uBAAIB,C;kBAA0B,sBAAS,mBAAO,kBAAuB,uBAAvB,  
C;kBAA8B,gB;;;kBAAxE,gB;;;gBADJ,gB;;;cAGI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAA  
U,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAy,uBAAZ,C;cAJX,gB;;;cAFJ,gB;  
;cASA,IAAI,iCAAJ,C;gBACI,gB;;;gBADJ,iB;;;cACI,IAAO,mBAAO,KAAd,IAAqB,uBAARb,C;gBAAA,gB;;;c  
ACI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;  
cAAA,Q;;cACA,mBAAO,qBAAy,uBAAZ,C;cAFX,gB;;;cAIA,IjL4K4C,CiL5KxC,mBjL4KyC,UiL5K7C,C;gBA  
AyB,iB;gCAA,iCAAM,mBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAAzB,iB;;;cAjCR,W;;cA4BI,iB;;c  
A1BJ,iB;;cAGI,KAAU,yBAAV,C;gBAAA,iB;;;6BAAU,sB;cACN,IAAI,kBAAO,CAAX,C;gBAAgB,oCAAQ,CA

AR,I;gBAAW,iB;;;gBAA3B,iB;;;;cACA,iBAAO,WAAI,YAAJ,C;cACP,IAAI,iBAAO,KAAP,KAAe,uBAAAnB,C;g  
BACI,iB;gCAAA,iCAAM,iBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,iB;;;;cAEI,IAAI,8BAAJ,C;gBA  
AiB,iBAAO,Q;;gBAAa,oBAAS,iBAAU,uBAAV,C;cAC9C,kBAAO,c;cAHX,iB;;;cAHJ,iB;;;;cASA,IjL+LgD,CiL/L  
5C,iBjL+L6C,UiL/LjD,C;gBACI,IAAI,qCAAKB,iBAAO,KAAP,KAAe,uBAArC,C;kBAA2C,iB;kCAAA,iCAAM,i  
BAAN,O;sBAAA,2C;2BAAA,yB;kBAAA,Q;;kBAA3C,iB;;;;gBADJ,iB;;;;cAdJ,W;;cAcI,iB;;;;cAZJ,iB;;;;cAkCJ,W;  
;K;IARCyB,sI;MAAA,yD;uBAAA,6K;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAF7B,6E;MACI,IAAI,CAA  
C,QAAS,UAAAd,C;QAAyB,OAAO,2B;MACHC,OAAO,WAAkB,0EAAIB,C;K;IAwCwB,6B;MAA8B,uB;MAA7B,  
kB;MAChC,mBAA6B,C;MAC7B,eAAyB,C;K;2CAEzB,8B;MACI,+DAAkB,SAAIB,EAA6B,OAA7B,EAA5C,W  
AAK,KAA3C,C;MACA,mBAAiB,S;MACjB,eAAa,UAAU,SAAV,I;K;0CAGjB,iB;MACI,+DAAkB,KAAIB,EAA  
yB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAy,KAAZ,IAAL,C;K;qFAGY,Y;MAAQ,mB;K;;IASR,wC;MAAqD,u  
B;MAApD,sB;MzKrDxB,IAAI,EyKuDQ,cAAc,CzKvDtB,CAAJ,C;QACI,cyKsD2B,wE;QzKrD3B,MAAM,gCAA  
yB,OAAQ,WAAjC,C;;MAFV,IAAI,EyKwDQ,cAAc,aAAO,OzKxD7B,CAAJ,C;QACI,gByKuDqC,wFAA+E,aAA  
O,O;QzKiD3H,MAAM,gCAAyB,SAAQ,WAAjC,C;;MyK2DV,kBAAuB,aAAO,O;MAC9B,oBAA8B,C;MAE9B,s  
BAAyB,U;K;kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;uCAGA,iB;MAGW,Q;MAFP,+DAAkB,KAAIB,  
EAAyB,SAAzB,C;MAEA,OAAO,sBAmGmC,CAnG5B,iBAmG6B,GAnGV,KAmGU,IAAD,IAAa,eAnGhD,4D;K  
;kCAGX,Y;MAAe,qBAAQ,e;K;IAEgB,4D;MAAA,sC;MAAS,2B;MAC5C,eAAoB,oB;MACpB,eAAoB,4B;K;8D  
AEpB,Y;MAKgB,Q;MAJZ,IAAI,iBAAS,CAAb,C;QACI,W;;QAGA,mBAAQ,sCAAO,YAAP,4DAAR,C;QACA,e  
AoFkC,CAPf1B,YAoF2B,GAPfB,CAoFa,IAAD,IAAa,+B;QANf/C,mC;;K;;oCAXZ,Y;MAAuC,kD;K;2CAGvC,i  
B;MAGiE,UAQ1C,MAROC,EAe1C,MAf0C,EAqBtD,M;MatBP,aACQ,KAAM,OAAN,GAAa,IAAK,KAAtB,GA  
AkC,UAAAN,KAAM,EAAO,IAAK,KAAZ,CAAIC,GAAyD,kD;MAE7D,WAAW,IAAK,K;MAEhB,WAAW,C;MA  
CX,UAAU,iB;MAEV,OAAO,OAAO,IAAP,IAAe,MAAM,eAA5B,C;QACI,OAAO,IAAP,IAAe,wBAAO,GAAP,g  
E;QACf,mB;QACA,iB;;MAGJ,MAAM,C;MACN,OAAO,OAAO,IAAd,C;QACI,OAAO,IAAP,IAAe,wBAAO,GA  
AP,gE;QACf,mB;QACA,iB;;MAEJ,IAAI,MAAO,OAAP,GAAC,IAAK,KAAvB,C;QAA6B,OAAO,IAAK,KAAZ,I  
AAoB,I;MAEjD,OAAO,uD;K;mCAGX,Y;MACI,OAAO,qBAAQ,gBAAa,SAAb,OAAR,C;K;4CAGX,uB;MAKI,k  
BAAoD,eAAjC,mBAAy,mBAAa,CAAzB,IAA8B,CAA9B,IAAiC,EAAa,WAAb,C;MACpD,gBAAoB,sBAAC,CA  
AlB,GAA4B,UAAP,aAAO,EAAO,WAAp,CAA5B,GAAqD,qBAAQ,gBAAa,WAAb,OAAR,C;MACrE,OAAO,eA  
AW,SAAX,EAA5B,SAAtB,C;K;qCAGX,mB;MAII,IAAI,aAAJ,C;QACI,MAAM,6BAAsB,qBAAtB,C;;MAGV,cA  
6B0C,CA7BnC,iBA6BoC,GA7BjB,SA6BiB,IAAD,IAAa,eA7BvD,IAAmC,O;MACnC,6B;K;+CAGJ,a;MzKhJA,I  
AAI,EyKoJQ,KAAK,CzKpJb,CAAJ,C;QACI,cyKmJkB,wC;QzKIJIB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV  
,IAAI,EyKqJQ,KAAK,SzKrJb,CAAJ,C;QACI,gByKoJqB,wEAA8D,S;QzKnJnF,MAAM,gCAAyB,SAAQ,WAAjC  
,C;;MyKqJN,IAAI,IAAI,CAAR,C;QACI,YAAy,iB;QACZ,UAGBsC,CAhB5B,KAgB6B,GAhBf,CAGBe,IAAD,IA  
Aa,e;QAdnD,IAAI,QAAQ,GAAZ,C;UACW,OAAP,aAAO,EAAK,IAAL,EAAW,KAAx,EAAkB,eAAIB,C;UACA,  
OAAP,aAAO,EAAK,IAAL,EAAW,CAAX,EAAc,GAAd,C;;UAEA,OAAP,aAAO,EAAK,IAAL,EAAW,KAAx,E  
AAkB,GAAIB,C;;QAGX,oBAAa,G;QACb,wBAAQ,CAAR,I;K;qCAKR,wB;MAC8C,QAAC,YAAO,CAAP,IAA  
D,IAAa,e;K;;IA9G3D,0C;MAAA,oD;MAA6B,uBAAK,gBAAMb,QAANb,OAAL,EAAmC,CAAnC,C;MAA7B,Y;  
K;ICvFJ,0C;MAIL,QAAQ,I;MACR,QAAQ,K;MACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB  
,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OtL+B4E,0BsL/BrE,kBAAM,CAAN,CtL0Q2B,KAAL,GAAiB,G  
A308B,EsL/B1D,KtL0QgB,KAAL,GAAiB,GA308B,CsL/BrE,IAAP,C;UACI,a;;QACJ,OtL6B4E,0BsL7BrE,kBA  
AM,CAAN,CtLwQ2B,KAAL,GAAiB,GA308B,EsL7B1D,KtLwQgB,KAAL,GAAiB,GA308B,CsL7BrE,IAAP,C;  
UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,  
CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,uC;MA  
GI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;Q  
ACI,UAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,UAAU,  
KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAIL,QAAQ,I;MACR,QAAQ,K;MACR,YAAy,kBAAM  
,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OpLM6E,0BoLNtE,  
kBAAM,CAAN,CpL0O2B,KAAL,GAAiB,KApO+B,EoLN3D,KpL0OgB,KAAL,GAAiB,KApO+B,CoLNtE,IAAP  
,C;UACI,a;;QACJ,OpLI6E,0BoLJtE,kBAAM,CAAN,CpLwO2B,KAAL,GAAiB,KApO+B,EoLJ3D,KpLwOgB,KA  
AL,GAAiB,KApO+B,CoLJtE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;U



ACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UAC  
A,a;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,  
QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MA  
CJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;  
MACR,QAAQ,K;MACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,K  
AAK,CAAZ,C;QACI,OrLnB8D,YqLmBvD,kBAAM,CAAN,CrLnBwE,KAAjB,EqLmB5C,KrLnByE,KAA7B,CqL  
mBvD,IAAP,C;UACI,a;;QACJ,OrLrB8D,YqLqBvD,kBAAM,CAAN,CrLrBwE,KAAjB,EqLqB5C,KrLrByE,KAA  
7B,CqLqBvD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,C  
AAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OA  
AO,C;K;IAGX,yC;MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,C  
AAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,K  
AAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;M  
ACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QAC  
I,OrK5C+D,aqK4CxD,kBAAM,CAAN,CrK5C0E,KAAiB,EqK4C7C,KrK5C2E,KAA9B,CqK4CxD,IAAP,C;UACI  
,a;;QACJ,OrK9C+D,aqK8CxD,kBAAM,CAAN,CrK9C0E,KAAiB,EqK8C7C,KrK9C2E,KAA9B,CqK8CxD,IAAP,  
C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAA  
M,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,yC;  
MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,  
C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YA  
AU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAKR,gD;MAI6E,UAAU,KAAV,EAAiB,SAAjB,EAA4B,UAA  
U,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC  
7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,K  
AAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IxK9I7E,0C;MF0BI,IAAI,EEjBI,SAAU,OAAV,GAAiB,  
CFiBrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAyB,OAAQ,WAAjC,C;;MEIBV,OAAO,oBAAoB,CAApB,E  
AAuB,CAAvB,EAA0B,SAA1B,C;K;IAGX,8C;MACe,Q;MAAX,wBAAW,SAAX,gB;QAAW,SAAA,SAAX,M;Q  
ACI,SAAS,GAAG,CAAH,C;QACT,SAAS,GAAG,CAAH,C;QACT,WAAW,cAAc,EAAd,EAaKB,EAAlB,C;QAC  
X,IAAI,SAAQ,CAAZ,C;UAAe,OAAO,I;;MAE1B,OAAO,C;K;sGAGX,yB;MAAA,8D;MAAA,iC;QASI,OAAO,c  
AAc,SAAS,CAAT,CAAd,EAA2B,SAAS,CAAT,CAA3B,C;O;KATX,C;sGAYA,sC;MASI,OAAO,UAAW,SAAQ,  
SAAS,CAAT,CAAR,EAAqB,SAAS,CAAT,CAArB,C;K;IAAtB,6B;MAWY,Q;MALR,IAAI,MAAM,CAAV,C;QA  
Aa,OAAO,C;MACpB,IAAI,SAAJ,C;QAAe,OAAO,E;MActB,IAAI,SAAJ,C;QAAe,OAAO,C;MAGtB,OAA8B,iB  
AAtB,mDAAsB,EAAU,CAAV,C;K;IAaZ,6C;MAAA,uB;QAAU,2BAAoB,CAApB,EAAuB,CAAvB,EAA0B,iBA  
A1B,C;O;K;IAVhC,8B;MF7CI,IAAI,EEsDI,SAAU,OAAV,GAAiB,CfIdrB,CAAJ,C;QACI,cAda,qB;QAeb,MAA  
M,gCAAyB,OAAQ,WAAjC,C;;MEqDV,OAAO,eAAW,2BAAX,C;K;0FAIX,yB;MAAA,sC;MAAA,oC;MAAA,u  
BAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,E  
AA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MAPf,2B;QAOI,sBAAW,0BAAX,C;O;KAPJ,C;0FASA,yB;M  
AAA,oC;MAQe,gE;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC,OA1Dd,UAAW,SAAQ,SA0DW,C  
A1DX,CAAR,EAAqB,SA0DC,CA1DD,CAArB,C;S;O;MAkDtB,uC;QAQI,sBAAW,sCAAX,C;O;KARJ,C;4GAU  
A,yB;MAAA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/  
Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MAPf,2B;QAOI,sBAAW,o  
CAAX,C;O;KAPJ,C;8GASA,yB;MAAA,oC;MAUe,0E;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC  
,OA/Ed,UAAW,SAAQ,SA+EW,CA/EX,CAAR,EAAqB,SA+EC,CA/ED,CAArB,C;S;O;MAqEtB,uC;QAUI,sBAA  
W,gDAAX,C;O;KAVJ,C;kFAYA,yB;MAAA,sC;MAAA,oC;MAAA,oBAQe,yB;QA9Gf,8D;eA8Ge,yC;UAAA,uB;  
YACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;YACIC,Q;YAAA,IAAI,oBAAmB,CAAvB,C;cAAA,OA  
A0B,e;;cAAqB,eAAsB,gB;cAArE,OA+GG,cAAc,SAuG8C,CAvG9C,CAAd,EAA2B,SAuGoC,CAvGpC,CAA3B,  
C;;YAsGH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,kCAAX,C;O;KARJ,C;oFAaA,yB;MAAA,oC;MAQe,0E;Q  
AAA,uB;UACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;UACIC,Q;UAAA,IAAI,oBAAmB,CAAvB,C;Y  
AAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OA+GG,UAAW,SAAQ,SAwGyC,CAXGzC,CA  
AR,EAAqB,SAwG+B,CAXG/B,CAArB,C;;UAuGd,W;S;O;MATR,kD;QAQI,sBAAW,8CAAX,C;O;KARJ,C;sGAa

A,yB;MAAA,sC;MAAA,oC;MAAA,8BAQe,yB;QAxIf,8D;eAwIe,mD;UAAA,uB;YACP,sBAAsB,qBAAsB,SAA Q,CAAR,EAAW,CAAX,C;YAC5C,Q;YAAA,IAAI,oBAAmB,CAAvB,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cA ArE,OAjIG,cAAc,SaiI8C,CAjI9C,CAAd,EAA2B,SaiIoC,CAjIpC,CAA3B,C;;YAgIH,W;W;S;OADO,C;MARf,sC ;QAQI,sBAAW,4CAAX,C;O;KARJ,C;wGAaA,yB;MAAA,oC;MAQe,8F;QAAA,uB;UACP,sBAAsB,qBAAsB,SA AQ,CAAR,EAAW,CAAX,C;UAC5C,Q;UAAA,IAAI,oBAAmB,CAAvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,k B;YAAtB,eAAkC,gB;YAAjF,OAIIG,UAAW,SAAQ,SakIyC,CAIzC,CAAR,EAAqB,SakI+B,CAII/B,CAArB,C;; UAiId,W;S;O;MATR,kD;QAQI,sBAAW,wDAAX,C;O;KARJ,C;kGAcA,yB;MAAA,oC;MAOe,wE;QAAA,uB;UA CP,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;UAA1C,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+ C,mBAAW,CAAX,EAAc,CAAd,C;S;O;MATvD,wC;QAOI,sBAAW,4CAAX,C;O;KAPJ,C;IAmBe,oD;MAAA,uB ;QACP,sBAAsB,SAAU,SAAQ,CAAR,EAAW,CAAX,C;QAAhC,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA +C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,uC;MAOI,sBAAW,kCAAX,C;K;IAYc,wE;MAAA,uB;Q ACV,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;QAA1C,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GA A+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,+C;MAOI,sBAAc,4CAAd,C;K;IAaW,+C;MAAA,uB;Q AEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,E;aACb,c;UAHJ,OAGiB,C;;UAHjB,OAIY,kBAA W,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAZ/B,gC;MAOI,sBAAW,6BAAX,C;K;4FASJ,yB;MAAA,4D;MAAA,w D;MAAA,mB;QAOqE,kBAAW,cAAX,C;O;KAPrE,C;IAgBe,8C;MAAA,uB;QAEH,UAAM,CAAN,C;UADJ,OAC e,C;aACX,c;UAFJ,OAEiB,C;aACb,c;UAHJ,OAGiB,E;;UAHjB,OAIY,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O ;K;IAZ/B,+B;MAOI,sBAAW,4BAAX,C;K;0FASJ,yB;MAAA,4D;MAAA,sD;MAAA,mB;QAOoE,iBAAU,cAAV, C;O;KAPpE,C;IASA,wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F,wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F, gC;MAM+D,IAEJ,IAFI,EAGJ,M;MAFvD,kBAD2D,SAC3D,sB;QADqD,OAC5B,SAAK,W;WAC9B,WAF2D,SA E3D,wC;QAFqD,OAEe,4F;WACvD,WAH2D,SAG3D,wC;QAHqD,OAGE,gG;;QAHF,OAI7C,uBAAmB,SAAnB, C;K;IAIuB,wC;MAAC,4B;K;2CACChC,gB;MAAwC,OAAA,eAAW,SAAQ,CAAR,EAAW,CAAX,C;K;4CACnD,Y ;MACgC,sB;K;;IAGpC,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE, Y;MAC8C,2C;K;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;IAMA,kC;MAAA,sC;K;+CACI,gB;MAAoE,O AAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE,Y;MAC8C,2C;K;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC; K;8EyKjTA,4B;MAUI,OAAK,iBAAL,SAAK,EAAU,KAAs,C;K;ICTT,iC;K;;;oDayDI,0C;MAiB+D,oB;QAAA, 2C;aAjB/D,kG;K;;IAoBJ,uC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,qC;MAAA,wC;O;MASI,4E;MAMA,8E;M AOA,4E;MAOA,kE;K;;IApBA,mD;MAAA,2B;MAAA,2C;K;;IAMA,oD;MAAA,2B;MAAA,4C;K;;IAOA,mD;M AAA,2B;MAAA,2C;K;;IAOA,8C;MAAA,2B;MAAA,sC;K;;IA7BJ,iC;MAAA,+K;K;;IAAA,sC;MAAA,a;AAAA,c; UAAA,gD;aAAA,e;UAAA,iD;aAAA,c;UAAA,gD;aAAA,S;UAAA,2C;;UAAA,oE;;K;;oFAqCA,mB;K;;; ;I7HmBiD,gD;MAAA,oB;QACzC,WAAW,sBAAmB,YAAF,CAAE,CAAnB,C;QACX,cAAM,IAAN,C;QADA,O AEA,IAAK,a;O;K;;;IAtHb,+B;K;;iFAUA,yB;MAAA,4B;MAAA,mC;QAMI,6BDgDQ,WChDkB,KDgDIB,CChD R,C;O;KANJ,C;2GAQA,yB;MAAA,4B;MDgDQ,kD;MChDR,uC;QAOI,6BDgDQ,WAAO,cChDW,SDgDX,CAAP ,CChDR,C;O;KAPJ,C;+FAUA,yB;MAAA,kC;MAAA,mD;MAAA,yE;QASI,sC;QAAA,4C;O;MATJ,iGAWY,Y;Q AAQ,2B;OAXpB,E;MAAA,0DAaQ,kB;QACI,wBAAW,MAAX,C;O;MAZ,sF;MAAA,sC;QASI,0D;O;KATJ,C;I AiBA,gD;MAaI,4BAA0D,YAAzC,wCAA6B,UAA7B,CAAYC,CAA1D,EAAyE,yBAAzE,C;K;IAEJ,4D;MAcI,4B AAoE,YAAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAApE,EAAmF,yBAAAnF,C;K;IAEJ,+C;MAU6C,Y AAzC,wCAA6B,UAA7B,CAAYC,CAtEzC,oBDgDQ,WCSBsD,kBDtBtD,CChDR,C;K;IAyEJ,2D;MAWuD,YAAAn D,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CApFnD,oBDgDQ,WCoCgE,kBDpChE,CChDR,C;K;IAuFJ,+C;M AYI,OAA6C,8BAAtC,c;K;8EZX,yB;MAAA,oE;MAAA,6E;MAYiD,gD;QAAA,oB;UACzC,WAAW,sBAAmB, YAAF,CAAE,CAAnB,C;UACX,cAAM,IAAN,C;UADA,OAEA,IAAK,a;S;O;MAfb,sC;QAYW,mBAAAsC,8BAAt C,6B;QAAP,OAAO,kD;O;KAZX,C;qGA0BI,yB;MAAA,2D;MAAA,mB;QACI,MAAM,6BAAoB,0BAApB,C;O; KADV,C;;M8HzIA,yC;;IAAA,uC;MAAA,2C;K;;;IAAA,mD;MAAA,kD;QAAA,iC;;MAAA,2C;K;+EAKBA,wB;K ;oDAaA,e;MAK2C,IAAI,IAAJ,EAGK,M;MAL5C,IAAI,+CAAJ,C;QAEI,OAAW,GAAl,kBAAS,IAAK,IAAd,CA AR,GAA4B,cAAI,OAAJ,GAAl,iBAAQ,IAAR,CAAJ,yCAA5B,GAAyD,I;;MAGpE,OAAW,8CAA4B,GAAhC,GA AqC,8EAArC,GAAoD,I;K;yDAI/D,e;MAGI,IAAI,+CAAJ,C;QACI,OAAW,GAAl,kBAAS,IAAK,IAAd,CAAJ,IA A0B,GAAl,iBAAQ,IAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,I;;MAE/F,OAAW,8CAA4B,GAAhC,GAAq C,mCAArC,GAAgE,I;K;;;ICtChD,oD;MACf,cAAc,GAAl,kBAAS,OAAQ,IAAjB,C;MACIB,IAAI,YAAY,mCAA

hB,C;QADA,OACuC,O;;QAEnC,kBAaKB,oBAAQ,yCAAR,C;QACIB,IAAI,mBAAJ,C;UAJJ,OAI6B,oBAAgB,O  
AAhB,EAAYB,OAAzB,C;;UACrB,WAAW,OAAQ,kBAAS,yCAAT,C;UAL3B,OAMY,SAAS,mCAAb,GAAoC,o  
BAAgB,OAAhB,EAAYB,WAAzB,CAApC,GACI,oBAAgB,oBAAgB,IAAhB,EAAsB,OAAtB,CAAhB,EAAGD,W  
AAhD,C;;K;8CAdxB,mB;MAKI,OAAI,YAAY,mCAAhB,GAAuC,IAAvC,GACI,OAAQ,cAAK,IAAL,EAAY,4B  
AAX,C;K;;qDAiCz,e;MAEyB,Q;MADrB,OACI,OAAA,IAAK,IAAL,EAAY,GAAY,CAAJ,GAAqB,0EAArB,G  
AAoC,I;K;sDAExC,8B;MACI,iBAAU,OAAV,EAAMB,IAAnB,C;K;0DAEJ,e;MACI,OAAI,OAAA,IAAK,IAAL,E  
AAY,GAAY,CAAJ,GAAqB,mCAArB,GAAgD,I;K;;IC1DP,8C;MAAC,wB;K;kFAAA,Y;MAAA,yB;K;;IAiCe,wD  
;MAEjE,kC;MAEA,4BAAqC,mDAAJ,GAAkD,OAAQ,qBAA1D,GAA0E,O;K;4DAE3G,mB;MAA6C,+BAAS,OA  
AT,C;K;6DAC7C,e;MAA8C,eAAQ,IAAR,IAAgB,8BAAE,G;K;;IAGjF,+C;MAW2C,IAAI,IAAJ,EAGV,M;MAL7  
B,IAAI,+CAAJ,C;QAEI,OAAW,GAAL,kBAAS,SAAK,IAAd,CAAR,GAA4B,cAAI,OAAJ,GAAL,iBAAQ,SAAR,C  
AAJ,yCAA5B,GAAYD,I;;MAGpE,OAAW,SAAK,IAAL,KAAa,GAAjB,GAAsB,mFAAtB,GAAqC,I;K;IAGhD,6C  
;MAUI,IAAI,+CAAJ,C;QACI,OAAW,GAAL,kBAAS,SAAK,IAAd,CAAJ,IAA0B,GAAL,iBAAQ,SAAR,CAAJ,QA  
A9B,GAAYD,mCAAzD,GAAoF,S;;MAE/F,OAAW,SAAK,IAAL,KAAa,GAAjB,GAAsB,mCAAtB,GAAiD,S;K;I  
AG5D,iC;MAAA,qC;MAKI,4B;K;oDACA,Y;MAAiC,0C;K;kDAEjC,e;MAAYD,W;K;mDACzD,8B;MAA4E,c;K;  
mDAC5E,mB;MAAwE,c;K;uDACxE,e;MAA8D,W;K;+CAC9D,Y;MAAsC,Q;K;+CACtC,Y;MAAYC,8B;K;;IAb7  
C,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAqB8B,wC;MAC1B,kB;MACA,wB;K;4CAGA,e;MAGQ,Q;MAFJ,U  
AAU,I;MACV,OAAO,IAAP,C;QACI,YAAA,GAAL,UAAJ,aAAY,GAAY,W;UAAwB,W;;QACxB,WAAW,GAAL,  
O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,iBAAK,GAAL,C;;K;6CAKnB,8B;MACI,iBAAU,WAA  
K,cAAK,OAAL,EAAC,SAAd,CAAF,EAAYC,cAAzC,C;K;iDAEJ,e;UAGW,I;MAFP,+BAAQ,GAAR,U;QAAoB,O  
AAO,W;;MAC3B,cAAc,WAAK,kBAAS,GAAT,C;MAEf,gBAAY,WAAZ,C;QAAoB,W;WACpB,gBAAY,mCAA  
Z,C;QAAqC,qB;;QAC7B,2BAAgB,OAAhB,EAAYB,cAAzB,C;MAHZ,W;K;uCAOJ,Y;MAIc,IAAI,IAAJ,Q;MAH  
V,UAAU,I;MACV,WAAW,C;MACX,OAAO,IAAP,C;QACU,uBAAI,OAAJ,GAAL,OAAJ,gC;QAAA,mB;UAAgC  
,OAAO,I;;QAA7C,MAAM,M;QACN,mB;;K;2CAIR,mB;MACI,+BAAI,OAAQ,IAAZ,GAAoB,OAApB,C;K;8CA  
EJ,mB;MAQ4B,Q;MAPxB,UAAU,O;MACV,OAAO,IAAP,C;QACI,IAAI,CAAC,gBAAS,GAAL,UAAb,CAAL,C;  
UAA4B,OAAO,K;QACnC,WAAW,GAAL,O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,gBAAS,0EA  
AT,C;;K;uCAKnB,iB;MACI,gBAAS,KAAT,KAaKB,yCAA4B,KAAM,SAAN,KAAgB,aAA5C,IAAsD,KAAM,e  
AAY,IAAZ,CAA9E,C;K;yCAEJ,Y;MAA+B,OAAK,SAAL,WAAK,CAAL,GAA0B,SAAR,cAAQ,CAA1B,I;K;IA  
GZ,uD;MACX,OAAI,G5JyHoC,YAAU,C4JzHID,GAAMB,OAAQ,WAA3B,GAA6C,GAAL,UAAQ,O;K;yCAF3D  
,Y;MACI,aAAM,kBAAK,EAAL,EAAS,+BAAT,CAAN,GAEL,G;K;IAMO,8E;MAAA,6B;QAAyB,Q;QAAT,iBA  
AS,sBAAT,EAAS,8BAAT,UAAoB,O;QAAQ,W;O;K;+CAJ3D,Y;MAOsB,Q;MANIB,QAAQ,a;MACR,eAAe,gBA  
A+B,CAA/B,O;MACf,gBAAY,CAAZ,C;MACA,kBAAK,kBAAL,EAAY,oDAAX,C;M/KtFJ,IAAI,E+KuFM,YA  
AS,C/KvFf,CAAJ,C;QACI,cAdW,e;QAEX,MAAM,6BAAsB,OAAQ,WAA9B,C;;M+KuFN,OAAO,+BAAW,qDA  
AX,C;K;IAGa,8C;MACpB,kD;MADqB,wB;K;IACrB,gD;MAAA,oD;MACI,4B;K;;IADJ,4D;MAAA,2D;QAAA,0  
C;;MAAA,oD;K;yDAIA,Y;MAA0C,gBAAT,a;M7Lm9YrB,Q;MADhB,kB6Ll9YmD,mC;M7Lm9YnD,wBAAgB,S  
AAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAwB,yBAAa,OAAb,C;;M6Ln9YT,O7Lo9Y9B,W;K;;I8LtoZX,o  
E;MA4BI,MAAM,wBAAoB,sEAApB,C;K;8GA5BV,yB;MAAA,2D;MAAA,sC;QA4BI,MAAM,6BAAoB,sEAAp  
B,C;O;KA5BV,C;IA0CoC,mC;MAAQ,4D;K;IAE5C,4C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;  
O;MAK0C,oG;MAAqB,gF;MAAW,4E;K;IAAhC,+D;MAAA,gC;MAAA,uD;K;;IAAqB,qD;MAAA,gC;MAAA,6  
C;K;;IAAW,mD;MAAA,gC;MAAA,2C;K;;IAL1E,sC;MAAA,sJ;K;;IAAA,2C;MAAA,a;AAA,qB;UAAA,4D;aA  
AA,W;UAAA,kD;aAAA,S;UAAA,gD;;UAAA,qF;;K;6ECnDA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA  
3B,SAAL,GAAuB,KAAS,C;O;KAHvF,C;2EAKA,yB;MAAA,0B;MAAA,mC;QAGqD,OAAgC,OAA1B,SAAL,G  
AAsB,KAAS,C;O;KAHf,C;6EAKA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA3B,SAAL,GAAuB,KAAS,  
C;O;KAHvF,C;6EAKA,yB;MAAA,0B;MAAA,4B;QAGqC,OAAqB,OAAP,CAAR,SAAE,C;O;KAH1D,C;+EAMA  
,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;6EAKA,yB;MAAA,4  
B;MAAA,mC;QAGwD,OAAgC,QAA1B,SAAL,GAAsB,KAAS,C;O;KAHxF,C;+EAKA,yB;MAAA,4B;MAAA,m  
C;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;+EAKA,yB;MAAA,4B;MAAA,4B;QAGuC,O  
AAqB,QAAP,CAAR,SAAE,C;O;KAH5D,C;ICpCA,qC;K;;ICAA,mB;K;;IAOA,iB;K;;IAOA,2C;K;;IAOA,wB;K;;I  
AQA,0B;K;;IAOA,sB;K;;IAOA,4B;K;;IAOA,6C;K;;IA+BuC,wE;MAEnC,uB;QAAA,UAsB,E;MACTB,qB;QAA

A,8B;MACA,2B;QAAA,qE;MACA,yB;QAAA,YAAqB,E;MAJrB,sB;MACA,sB;MACA,kB;MACA,8B;MACA,0  
B;K;;IAGJ,iD;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,+C;MAAA,kD;O;MAKI,wG;MACA,wG;MACA,8F;K;;I  
AFA,iE;MAAA,qC;MAAA,yD;K;;IACA,iE;MAAA,qC;MAAA,yD;K;;IACA,4D;MAAA,qC;MAAA,oD;K;;IAPJ,2  
C;MAAA,6K;K;;IAAA,gD;MAAA,a;aAAA,kB;UAAA,8D;aAAA,kB;UAAA,8D;aAAA,a;UAAA,yD;;UAAA,6E;;  
K;;IAUA,wB;K;;ICjGA,qB;MAAA,yB;K;0CAII,Y;MAO6D,uB;K;2HAE7D,yB;MAAA,+D;MAAA,kC;MAAA,0F  
;MAAA,6F;MAAA,4E;QAUI,wC;QAAS,2C;O;MAVb,mEAWQ,wC;QAA6E,sBAAS,QAAT,EAAmB,QAAnB,EA  
A6B,QAA7B,C;O;MAXrF,oG;MAAA,yC;QAUI,wDAA+B,YAA/B,C;O;KAVJ,C;uHAcA,yB;MAAA,+D;MAAA,  
kC;MAAA,wF;MAAA,yF;MAAA,0E;QAcI,wC;QAAS,2C;O;MADB,kEAeQ,wC;QAAuF,6BAAS,QAAT,EAAmB,  
QAAnB,EAA6B,QAA7B,C;O;MAf/F,kG;MAAA,yC;QAcI,sDAA+B,YAA/B,C;O;KADJ,C;;;IA3BJ,iC;MAAA,gC;  
QAAA,e;;MAAA,yB;K;IAGDiC,sB;MAC7B,eAAwB,I;K;4CAExB,6B;MACW,Q;MAAA,mB;MAAA,iB;QAAS,M  
AAM,6BAAsB,cAAY,QAAS,aAArB,uCAAtB,C;;MAAtB,OAAO,I;K;4CAGX,oC;MACI,eAAa,K;K;;;;kDC9CjB,  
6B;;K;;;;;;iEA+CA,6B;;K;;ICrDuC,0C;MACvC,uBAAoB,Y;K;wDAEpB,wC;MAM6F,W;K;uDAE7F,wC;K;oDA  
MA,6B;MACI,OAAO,oB;K;oDAGX,oC;MACI,eAAe,IAAK,gB;MACpB,IAAI,CAAC,0BAAa,QAAb,EAAuB,QA  
AvB,EAAiC,KAAjC,CAAL,C;QACI,M;;MAEJ,uBAAa,K;MACb,yBAAy,QAaz,EAAsB,QAAtB,EAAgC,KAAh  
C,C;K;;4EC9BR,wC;MAqBI,OAAO,e;K;4EAGX,+C;MAuBI,cAAI,KAAJ,C;K;4EAIJ,wC;MAMBI,OAAO,cAAI,  
OAAJ,C;K;4EAGX,+C;MAqBI,cAAI,OAAJ,EAAa,KAAb,C;K;IC/FJ,kB;MA6PI,4B;K;+BAtoA,Y;MAOiC,6BAA  
S,EAAT,C;K;uCAEjC,iB;MAW2C,4BAAQ,CAAR,EAaw,KAAX,C;K;uCAE3C,uB;MAakB,Q;MAHd,iBAaiB,I  
AAjB,EAAuB,KAAvB,C;MACA,QAAQ,QAAQ,IAAR,I;MACR,IAAI,IAAI,CAAJ,IAAS,MAAK,WAAIB,C;QAC  
c,IAAI,MAAM,CAAC,CAAD,IAAN,OAAy,CAAhB,C;UACN,eAAe,SAAS,CAAT,C;UAcF,6BAAS,QAAT,C;;U  
AEA,K;;YAEI,WAAW,cAAU,KAAK,C;YAC1B,IAAI,OAAO,C;;UACN,gBAAO,CAAP,IAAY,CAAZ,GAAGB,C  
AAhB,SAAqB,CAArB,C;UACT,Q;;QATJ,c;QAWA,OAAO,OAAO,GAAP,I;;QAEP,OAAO,IAAP,C;UACI,YAA  
U,c;UACV,IAAW,IAAP,qBAakB,KAAtB,C;YAA6B,OAAO,K;;K;gCAKhD,Y;MAOmC,OAAU,oBAAV,cAAU,  
CAAS,WAAI,EAAJ,CAAnB,yBAA6B,cAA7B,E;K;wCAEnC,iB;MAW8C,iCAAY,KAAZ,C;K;wCAE9C,uB;MAi  
BkB,Q;MAPd,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,eAAQ,IAAR,C;MACR,IAAI,eAAI,CAAR,C;Q  
ACI,O;QACA,IAAI,aAAO,CAAD,aAAN,GAAY,CAAZ,CAAJ,C;UACI,WAAW,CAAE,Q;UACb,YAAa,qBAAO,  
EAAP,CAAW,Q;UAEpB,aAAQ,CAAR,C;YACI,eAAe,SAAS,IAAT,C;YAEf,OAAmB,oBAAAnB,sBAAS,QAAT,C  
AAmB,CAAnB,iB;iBAEJ,cAAS,CAAT,C;YAEI,OAAU,oBAAV,cAAU,CAAV,iB;;YAEA,iBAae,SAAS,KAAT,  
C;YACf,OAAmB,oBAAAnB,sBAAS,UAAT,CAAmB,CAAS,WAAI,EAAJ,CAA5B,KAAiD,oBAAV,cAAU,CAAV,  
iBAAvC,C;;UAXR,U;;UAeA,K;;YAEI,WAAW,eAAW,oBAAK,CAAL,C;YActB,IAAI,YAAO,CAAP,C;;UACC,s  
BAAO,CAAP,MAAY,+BAAI,CAAJ,EAaz,eAAqB,CAArB,C;UACT,MAAM,C;;QAEV,OAAO,SAAO,GAAP,C;  
;QAEP,OAAO,IAAP,C;UACI,YAAU,e;UACV,IAAW,IAAP,0CAakB,KAAIB,CAAJ,C;YAA6B,OAAO,K;;K;m  
CAKhD,Y;MAKyC,6BAAS,CAAT,MAAe,C;K;kCAExD,Y;MAKuC,uBAAgB,sBAAS,EAAT,CAAhB,EAA8B,sB  
AAS,EAAT,CAA9B,C;K;0CAEvC,iB;MASoD,+BAAW,GAAX,EAAGB,KAAhB,C;K;0CAEpD,uB;MAcY,Q;MA  
FR,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,WAAW,QAAQ,I;MACX,IAAS,WAAI,IAAK,CAAL,IAA0B,SA  
AL,IAAK,CAA1B,IAA8C,SAAN,KAAM,CAAI,D,C;QACJ,SAAS,qBAAgB,QAAQ,CAAR,GAAY,OAAO,CAAn  
C,C;QACT,cAAO,EAAP,GAAY,E;;QAEZ,cAAO,oBAAe,I;;MAJ1B,Y;MAMA,OAAW,KAAK,KAAT,GAAsB,S  
AAN,KAAM,CAAtB,GAAsC,C;K;iCAGjD,Y;MAKqC,6BAAS,EAAT,IAA0B,Q;K;IAWK,oF;MAAA,mB;QAAE,  
uBAAa,iBAAb,sBAAqC,eAArC,+BAAqE,aAAM,OAA3E,M;O;K;iDATtE,qC;MxLjLA,IAAI,EwL0LqB,CAAb,8  
BAAgB,KAAM,OxL1L9B,GwL0LiD,CAAX,0BAAc,KAAM,OxL1L1D,GwL0LsC,KxL1LtC,CAAJ,C;QACI,cwL  
yLgE,kDxLzLID,E;QACd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EwL2LQ,aAAa,OxL3LrB,CAAJ,C;  
QACI,gBwL0LgC,mF;QxLzLhC,MAAM,gCAAyB,SAAQ,WAAjC,C;;MwL2LN,YAAy,CAAC,UAAU,SAAV,IA  
AD,IAAwB,CAAxB,I;MAEZ,mBAAe,SAAf,C;MrLzEJ,iBAAc,CAAd,UqL0EW,KrL1EX,U;QqL2EQ,QAAQ,c;Q  
ACR,MAAM,UAAN,IAAoB,OAAf,CAAE,C;QACpB,MAAM,aAAW,CAAX,IAAN,IAAgC,OAAV,CAAE,KAA  
K,CAAG,C;QAChC,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAx,CAAE,KAAK,EAAL,C;QACjC,MAAM,aAAW,  
CAAX,IAAN,IAAiC,OAAx,CAAE,KAAK,EAAL,C;QACjC,0BAAy,CAAZ,I;;MAGJ,gBAAgB,UAAU,UAAV,I;  
MACHB,SAAS,sBAAS,YAAy,CAAZ,IAAT,C;MACT,aAAU,CAAV,MAakB,SAIb,M;QACI,MAAM,aAAW,C  
AAX,IAAN,IAAQ,C,OAAf,EAAG,MAAK,IAAI,CAAJ,IAAL,CAAY,C;;MAGzC,OAAO,K;K;yCACX,uD;MAvB4  
C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,O;aARrF,0H;K;yCAiCA,iB;MAOyD,8BAAU,KAAV,E

AAiB,CAAjB,EAAoB,KAAM,OAA1B,C;K;yCAEzD,gB;MAKkD,8BAAU,cAAU,IAAV,CAAV,C;K;IAGID,0B;  
MAAA,8B;MAO2B,iB;MACvB,uBAAoC,uB;K;IAEpC,qC;MAAA,yC;MACI,4B;K;wDAEA,Y;MAAiC,mC;K;;I  
AHRc,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;8CAMA,Y;MAAkC,8C;K;gDAEIC,oB;MAA4C,OAAA,oBAAc,k  
BAAS,QAAT,C;K;uCAC1D,Y;MAA8B,OAAA,oBAAc,U;K;+CAC5C,iB;MAAwC,OAAA,oBAAc,iBAAQ,KAA  
R,C;K;+CACtD,uB;MAAmD,OAAA,oBAAc,iBAAQ,IAAR,EAAc,KAAc,C;K;wCAEjE,Y;MAAgC,OAAA,oBAA  
c,W;K;gDAC9C,iB;MAA2C,OAAA,oBAAc,kBAAS,KAAT,C;K;gDACzD,uB;MAAuD,OAAA,oBAAc,kBAAS,I  
AAT,EAAe,KAAf,C;K;2CAErE,Y;MAAsC,OAAA,oBAAc,c;K;0CAEpD,Y;MAAoC,OAAA,oBAAc,a;K;kDACID  
,iB;MAAiD,OAAA,oBAAc,oBAAW,KAAx,C;K;kDAC/D,uB;MAA+D,OAAA,oBAAc,oBAAW,IAAX,EAAiB,K  
AAjB,C;K;yCAE7E,Y;MAAkC,OAAA,oBAAc,Y;K;iDAEhD,iB;MAAsD,OAAA,oBAAc,mBAAU,KAAV,C;K;iD  
ACpE,gB;MAA+C,OAAA,oBAAc,mBAAU,IAAV,C;K;yDAC7D,qC;MACI,OAAA,oBAAc,mBAAU,KAAV,EA  
AiB,SAAjB,EAA4B,OAA5B,C;K;;IAtCtB,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IAOcj,wB;MAAuC,yBAAa,I  
AAb,EAAmB,IAAK,IAAI,EAA5B,C;K;IAEvC,wB;MAawC,yBAAa,IAAK,QAA1B,EAA2B,IAAK,YAAI,EAAJ,C  
AAQ,QAAxC,C;K;IAGxC,mC;MAUI,IAAA,KAAM,UAAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C;  
WACzB,IAAA,KAAM,KAAN,GAAa,UAAb,C;QAF8C,OAEhB,0BAAQ,KAAM,MAAd,EAAqB,KAAM,KAAN,  
GAAa,CAAb,IAArB,C;WAC9B,IAAA,KAAM,MAAN,GAAC,WAAd,C;QAH8C,OAGf,0BAAQ,KAAM,MAAN,  
GAAC,CAAd,IAAR,EAAyB,KAAM,KAA/B,IAAuC,CAAvC,I;;QAHe,OAIc,mB;K;IAGZ,oC;MAUI,IAAA,KAA  
M,UAAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KAAN,+C;QAFiD,OAEiB,2  
BAAS,KAAM,MAAf,EAA5B,KAAM,KAAN,yBAAa,CAAb,EAAtB,C;WAC/B,IAAA,KAAM,MAAN,+C;QAHi  
D,OAGjB,2BAAS,KAAM,MAAN,8BAAC,CAAd,EAAT,EAA0B,KAAM,KAAhC,0BAAwC,CAAxC,E;;QAHiB,  
OAIzC,oB;K;IAOZ,yB;MAAyC,YjFrTkB,MAAO,OiFqTpB,KjFrToB,CiFqTzB,I;K;IAEzC,4C;MAEI,OAAA,SAA  
K,KAAK,EAAL,GAAU,QAAf,GAAyC,CAAX,CAAC,QAAD,IAAW,KAAI,E;K;IAEjD,uC;MxLTVI,IAAI,EwLsV  
uD,QAAQ,IxLtV/D,CAAJ,C;QACI,cwLqVuE,+B;QxLpVvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IwLqVd,yC;  
MxLvVI,IAAI,EwLuVyD,sBAAQ,IAAR,KxLvVzD,CAAJ,C;QACI,cwLsVyE,+B;QxLrVzE,MAAM,gCAAYB,OA  
AQ,WAAjC,C;;K;IwLsVd,yC;MxLxVI,IAAI,EwLwV6D,QAAQ,IxLxVrE,CAAJ,C;QACI,cwLwV6E,+B;QxLTV7E  
,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IwLwVd,yC;MAAyD,oCAA0B,IAA1B,qBAAiC,KAAjC,kB;K;ICrXzD,6  
B;MAOqC,OpMmYE,SoMnYF,mBpMmYE,C;K;IoMjYvC,sC;MASgD,6BAAS,WAAT,EAAa,KAAb,C;K;IAEhD  
,4C;MAUI,qBAAqB,IAArB,EAA2B,KAA3B,C;MAEA,iBAAiB,IpMqQgB,KoMrQhB,GAAiB,W;MACiC,kBAAk  
B,KpMoQe,KoMpQf,GAakB,W;MAEpC,mBAAmB,0BAAQ,UAAAR,EAAoB,WAApB,IAAqC,W;MACxD,OpMs  
WmC,SoMtW5B,YpMsW4B,C;K;IoMnWvC,sC;MAWI,IAAA,KAAM,UAAAN,C;QAAMB,MAAM,gCAAYB,uCA  
AoC,KAA7D,C;;QACzB,IpMGkE,YoMHIE,KAAM,KpMG6E,KAAjB,EoMHRD,4BAAK,UpMG6E,KAA7B,CoM  
HIE,K;UAFiD,OAEiB,sBAAS,KAAM,MAAf,EpMqBsB,SoMrBA,KAAM,KpMqBI,KAAK,GAAW,CoMrBb,Wp  
MqBa,MAAX,IAAf,CoMrBtB,C;;UAC/B,IpMEkE,YoMFIE,KAAM,MpME6E,KAAjB,EoMFPD,4BAAK,UpME4  
E,KAA7B,CoMFIE,K;YAHiD,OpMuBI,SoMpBrB,sBpMiCsB,SoMjCb,KAAM,MpMiCiB,KAAK,GAAy,CoMjC1  
B,WpMiC0B,MAAZ,IAAf,CoMjCtB,EAA2B,KAAM,KAAjC,CpMoB+B,KAAK,GAAW,CoMpBN,WpMoBM,M  
AAX,IAAf,C;;YoMvBJ,OAIzC,mB;;K;IAGZ,8B;MAOuC,OpL0VG,UoL1VH,oBpL0VG,C;K;IoLxV1C,uC;MAS  
mD,8BAAU,2BAAV,EAAe,KAAf,C;K;IAEnD,6C;MAUI,sBAAsB,IAAtB,EAA4B,KAA5B,C;MAEA,iBAAiB,Ip  
LwNkB,KoLxNIB,8B;MACjB,kBAAkB,KpLuNiB,KoLvNjB,8B;MAEiB,mBAAmB,2BAAS,UAAAT,EAAqB,WA  
ArB,+B;MACnB,OpL6TsC,UoL7T/B,YpL6T+B,C;K;IoL1T1C,uC;MAWI,IAAA,KAAM,UAAAN,C;QAAMB,MA  
AM,gCAAYB,uCAAoC,KAA7D,C;;QACzB,IpL7CmE,aoL6CnE,KAAM,KpL7C+E,KAAIB,EoL6CtD,6BAAM,Up  
L7C8E,KAA9B,CoL6CnE,K;UAFoD,OAEpB,uBAAU,KAAM,MAAhB,EpLhCuB,UoLgCA,KAAM,KpLhCK,KA  
AK,KAAW,ChBsQ7C,UAAW,oBAAL,CoMtOyB,WpMsOzB,MAAK,CAAL,iBAAN,CgBtQ6C,MAAX,CAAhB,  
CoLgCvB,C;;UACHC,IpL9CmE,aoL8CnE,KAAM,MpL9C+E,KAAIB,EoL8CrD,6BAAM,UpL9C6E,KAA9B,CoL  
8CnE,K;YAHoD,OpL9BG,UoLiCtB,uBpLpBuB,UoLoBb,KAAM,MpLpBkB,KAAK,UAAy,ChByP/C,UAAW,oB  
AAL,CoMrOc,WpMqOd,MAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,CoLoBvB,EAA4B,KAAM,KAAIC,Cp  
LjCiC,KAAK,KAAW,ChBsQ7C,UAAW,oBAAL,CoMrOgC,WpMqOhC,MAAK,CAAL,iBAAN,CgBtQ6C,MAA  
X,CAAhB,C;;YoL8BH,OAI5C,oB;;K;IAGZ,sC;MAQI,4BAAU,KjK4+FH,QiK5+FP,C;MACA,OAAO,K;K;IAGX  
,uC;MAKsD,OjK2iG3C,eiK3iG2C,4BAAU,IAAV,CjK2iG3C,C;K;IiKziGX,4D;MAOGD,yB;QAAA,YAAiB,C;M  
AAG,uB;QAAA,UAAe,KAAM,K;MACrF,4BAAU,KjKy9FH,QiKz9FP,EAA+B,SAA/B,EAA0C,OAA1C,C;MAC

A,OAAO,K;K;IAIX,2C;MzLrHI,IAAI,EX2B8D,YoM0FD,KpM1FkB,KAAjB,EoM0FO,IpM1FsB,KAA7B,CoM0F  
D,IzLrH7D,CAAJ,C;QACI,cyLoH6E,+B;QzLnH7E,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IyLoHd,4C;MzLrHI,I  
AAI,EKMc+D,aoLmFC,KpLnFiB,KAAiB,EoLmFS,IpLnFqB,KAA9B,CoLmFC,IzLrHhE,CAAJ,C;QACI,cyLqHg  
F,+B;QzLpHhF,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;I0LpBc,6C;MAcxB,oC;MA/BA,iB;MANA,Y;MACA,Y;  
MACA,Y;MACA,Y;MACA,Y;MACA,sB;M1LYA,IAAI,E0LLQ,CAAC,WAAK,QAAL,GAAU,QAAV,GAAe,QA  
Af,GAAoB,QAARb,MAA2B,C1LKnC,CAAJ,C;QACI,c0LNwC,wD;QILOxC,MAAM,gCAAYB,OAAQ,WAAjC,  
C;;MGoHV,iBAAc,CAAd,UuLxHW,EvLwHX,U;QuLxHiB,c;;K;qCAGjB,Y;MAGI,QAAQ,Q;MACR,IAAI,IAAO,  
MAAO,C;MACIB,WAAI,Q;MACJ,WAAI,Q;MACJ,WAAI,Q;MACJ,SAAS,Q;MACT,WAAI,E;MACJ,IAAK,IAA  
O,KAAM,CAAd,GAAsB,EAAtB,GAA8B,MAAO,C;MACzC,WAAI,C;MACJ,gCAAU,MAAV,I;MACA,OAAO,I  
AAI,aAAJ,I;K;8CAGX,oB;MACI,OAAU,cAAV,cAAU,EAAC,QAAc,C;K;IAEd,kC;MAAA,sC;MACI,4B;K;;IAD  
J,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7BA,gD;MAAA,sD;MACQ,yBAAK,KAAL,EAAY,KAAZ,EAAMb,  
CAAnB,EAAsB,CAAtB,EAA+B,CAAN,KAAzB,EAAuC,SAAU,EAAX,GAAoB,UAAW,CAAR,E,C;MADR,Y;K;I  
CbiD,8C;MACjD,4B;MACA,0C;K;oEADA,Y;MAAA,2B;K;2EACA,Y;MAAA,kC;K;uCAGA,iB;MACI,OAAO,0  
CAAgC,kBAAa,KAAM,UAAAnB,KAC/B,mBAAS,KAAM,MAAf,KAAwB,0BAAgB,KAAM,aAAtB,CADO,CAA  
hC,C;K;yCAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAW,SAAN,UAAAM,CAAX,QAAqC,SAAb,iBA  
Aa,CAAR,C,I;K;yCAGIC,Y;MAAkC,OAAE,UAAf,qBAAU,iB;K;;IAGhD,kC;MAM6E,2BAAgB,SAAhB,EAAsB,I  
AAtB,C;K;;0DAYzE,iB;MAA2C,qCAAiB,UAAjB,EAAwB,KAAxB,KAAkC,8BAAiB,KAAjB,EAAwB,iBAAXB,  
C;K;iDAC7E,Y;MAAkC,QAAC,8BAAiB,UAAjB,EAAwB,iBAAXB,C;K;;IACr,gD;MAI3B,gBAAqB,K;MACrB,u  
BAA4B,Y;K;0FACD,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;2DAE1C,gB;MAA+D,YAAK,C;K;mDAEpE,iB;M  
AAgD,gBAAS,aAAT,IAAmB,SAAS,oB;K;0CAC5E,Y;MAAkC,SAAE,iBAAU,oBAAZ,C;K;yCAEIC,iB;MACI,O  
AAO,4CAA+B,kBAAa,KAAM,UAAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADnB,CA  
A/B,C;K;2CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAA  
c,CAAvC,I;K;2CAGIC,Y;MAAkC,OAAE,aAAf,qBAAW,oB;K;;IAGjD,oC;MAOqF,6BAAkB,SAIb,EAAwB,IA  
AxB,C;K;IAQvD,+C;MAI1B,gBAAqB,K;MACrB,uBAA4B,Y;K;yFACF,Y;MAAQ,oB;K;gGACD,Y;MAAQ,2B;  
K;0DAEzC,gB;MAA6D,YAAK,C;K;kDAEIE,iB;MAA+C,gBAAS,aAAT,IAAmB,SAAS,oB;K;yCAC3E,Y;MAAk  
C,SAAE,iBAAU,oBAAZ,C;K;wCAEIC,iB;MACI,OAAO,2CAA8B,kBAAa,KAAM,UAAAnB,KAC7B,kBAAU,KA  
AM,SAAhB,IAA0B,yBAAiB,KAAM,gBADpB,CAA9B,C;K;0CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAu  
B,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAC,CAAvC,I;K;0CAGIC,Y;MAAkC,OAAE,aAAf,qBAAW,o  
B;K;;IAGjD,oC;MAOkF,4BAAiB,SAAjB,EAAuB,IAAvB,C;K;oFAGIF,8B;MAQI,0BAAmB,2BAAS,OAAT,C;K;  
IAGvB,+C;MACI,IAAI,CAAC,UAAAL,C;QAAiB,MAAM,gCAAYB,iCAA8B,IAA9B,iBAAzB,C;K;IC5I3B,gC;MA  
cW,Q;MADP,IAAI,CAAC,6BAAW,KAAX,CAAL,C;QAAwB,MAAM,uBAAmB,sC/EjBzC,oB+EIByC,CAAnB,C  
;;MAC9B,OAAO,sD;K;IAMX,oC;MAAkC,Q;MAA9B,OAAW,6BAAW,KAAX,CAAJ,GAAuB,sDAAvB,GAAuC,  
I;K;;;ICvBhB,yC;MA2B9B,uC;MA1BA,wB;MAIA,gB;M7LQA,IAAI,E6LDS,iBAAY,IAAb,MAAuB,iBAAvB,  
C7LCR,CAAJ,C;QACI,c6LDQ,iBAAY,IAAhB,GACI,8CADJ,GAGI,sCAA0B,aAA1B,qC;Q7LDR,MAAM,gCAA  
yB,OAAQ,WAAjC,C;;K;yC6LKV,Y;MAAwC,Q;MAAA,oB;MACpC,iB;QAD8B,OACtB,G;WACR,oD;QAF8B,O  
AEF,SAAL,SAAK,C;WAC5B,6C;QAH8B,OAGd,iBAAK,SAAL,C;WACHB,8C;QAJ8B,OAIb,kBAAM,SAAN,C;;  
QAJa,mC;K;IAOIC,qC;MAAA,yC;MACI,YAGqC,oBAAgB,IAAhB,EAAsB,IAAtB,C;K;iGAQJ,Y;MAAQ,gB;K;4  
DAEzC,gB;MAOI,8DAAqC,IAArC,C;K;gEA EJ,gB;MAMI,uDAA8B,IAA9B,C;K;4DAEJ,gB;MAMI,wDAA+B,I  
AA/B,C;K;;IArCR,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;2CArCJ,Y;MAWI,oB;K;2CAXJ,Y;MAeI,gB;K;6CA  
fJ,0B;MAAA,2BAWI,8CAXJ,EAeI,kCAfJ,C;K;yCAAA,Y;MAAA,c;MAWI,yD;MAIA,qD;MAfJ,a;K;uCAAA,iB;  
MAAA,4IAWI,4CAXJ,IAeI,oCAfJ,I;K;ICLA,kC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,gC;MAAA,mC;O;MA  
YI,4D;MAKA,8C;MAKA,gD;K;;IAVA,2C;MAAA,sB;MAAA,mC;K;;IAKA,oC;MAAA,sB;MAAA,4B;K;;IAKA,  
qC;MAAA,sB;MAAA,6B;K;;IAtBJ,4B;MAAA,mG;K;;IAAA,iC;MAAA,a;MAAA,W;UAAA,wC;aAAA,I;UAAA,i  
C;aAAA,K;UAAA,kC;;UAAA,6D;;K;;6ECAA,yB;MAAA,4F;MAAA,2B;QASI,MAAM,mCAA8B,0EAA9B,C;O;  
KATV,C;ICkCA,+D;MAaW,Q;MAAP,OAAO,8CAA0,KAAP,EAAC,UAAAd,EAA0B,QAA1B,oC;K;IAGX,kC;MA  
liB,Q;MAAb,wBAAa,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;mFAGX,qB  
;MAGwD,gCAA0,EAAP,C;K;qFAExD,4B;MAG4E,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAOx  
D,4B;MAGmE,OAAA,yBAAO,KAAP,CAVX,gBAAO,EAAP,C;K;IAxD,wD;MAEQ,sB;QAAqB,yBAAO,UAA

U,OAAV,CAAP,C;WACrB,sD;QAA4B,yBAAO,OAAP,C;WAC5B,2B;QAAMb,yBAAO,kBAAP,C;;QACX,yBA Ae,SAAR,OAAQ,CAAf,C;K;IIL7EhB,+B;MAY6B,kBAAIB,QAAQ,SAAR,EAAC,EAAd,C;MACH,IX0EE,WWIE E,GAAK,CAAT,C;QAAY,MAAM,gCAAYb,oEAAzB,C;MADtB,OX4EO,W;K;IWvEX,wC;MAGbW,Q;MAAA,q CAAiB,KAAjB,C;MAAA,iB;QAA2B,MAAM,gCAAYb,8BAAO,SAAP,4CAA+C,KAAxE,C;;MAAXC,OAAO,I;K ;IAGX,qC;MAY6B,kBAAIB,QAAQ,SAAR,EAAC,EAAd,C;MAAP,OXmEqB,WWnEa,IAAM,CXmEjC,GAAqB, WAArB,GAA+B,I;K;IWhE1C,8C;MAGbI,WAAW,KAAx,C;MAC4B,kBAArB,QAAQ,SAAR,EAAC,KAAd,C;M AAP,OX+CqB,WW/CgB,IAAM,CX+CpC,GAAqB,WAArB,GAA+B,I;K;IW5C1C,gC;MAWI,IAAY,CAAR,8BA AW,CAAF,C;QACI,OAAO,YAAM,SAAN,C;;MAEX,MAAM,gCAAYb,SAAM,SAAN,4BAAzB,C;K;IAGV,yC;M AkBW,Q;MANP,IAAI,EAAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYb,oBAAiB,KAAjB,4CAAzB, C;;MAEV,IAAI,YAAO,CAAP,IAAY,aAAQ,KAAxB,C;QACI,MAAM,gCAAYb,WAAQ,SAAR,mDAAwD,KAAj F,C;;MAEH,IAAI,YAAO,EAAX,C;QACH,mBAAM,SAAN,C;;QAEA,0BAAM,SAAN,IAAa,EAAb,C;;MAHJ,W; K;IAuFJ,8B;MAWSc,+B;K;0EAEtC,4B;MAM8D,OAAK,oBAAL,SAAK,CAAL,GAakB,K;K;IAEHf,gD;MAQoC ,0B;QAAA,aAAsB,K;MACtD,IAAI,cAAQ,KAAZ,C;QAAMb,OAAO,I;MAC1B,IAAI,CAAC,UAAAL,C;QAAiB,O AAO,K;MAExB,gBAAqB,cAAL,SAAK,C;MACrB,iBAAuB,cAAN,KAAM,C;MAEHb,yBAAa,U;MAAb,U;QAA 2B,OfRMyB,oBEqMzB,SFrMyB,CAAY,cAfrB,YAAY,CAAZ,CEoNhB,KFrMyB,oBEqMI,UFrMJ,CAAY,cAfrB, YAAY,CAAZ,C;;MEoNID,W;K;IAGJ,gC;MAGyC,QAAQ,cAAA,sCAAK,cAAL,EAAoB,sCAAK,cAAzB,CAAR, 6B;K;ImL3OzC,6C;MAc6B,4B;QAAA,eAAuB,G;MACHd,wCAAsB,EAAtB,EAA0B,YAA1B,C;K;IAEJ,mE;MA KwC,yB;QAAA,YAAoB,E;MAAI,4B;QAAA,eAAuB,G;MjMGnF,IAAI,CmBwR+C,CAAC,Q8K1R5C,Y9K0R4C, CnBxRpD,C;QACI,ciMHiC,wC;QjMjC,MAAM,gCAAYb,OAAQ,WAAjC,C;;MiMHV,cAAY,gB;MAEC,yBAAS, mBAAS,YAAA,SAAU,OAAV,EAAMb,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAAkB,SAAlB,C;MA0E9D, gBAAGb,iBA1ET,OA0ES,C;M3Lg7CT,kBAAoB,gB;MAoSd,gB;MADb,YAAY,C;MACC,O2L9xDN,O3L8xDM, W;kBAAb,OAAa,cAAb,C;QAAA,sB;QA1RsB,U;QAAA,cA0RT,oBAAMb,cAAnB,EAAMb,sBAAnB,U;Q2L/sDl B,kB;;YAHA,CAAC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,I2LltDH,C;UAC5C,a;;UAEA,4B;UA9E+B,u B;;Y/KgHzB,kC;YAAA,wBZ6qDyC,IY7qDzC,C;YAAA,qB;YAAA,oB;YAAA,oB;YAAAd,gE;cACI,I+KjHkD,CA Al,aAAH,U/KiHrC,YZ4qDqC,IY5qDrC,YAAK,OAAL,E+KjHqC,CAAG,C/KiHtD,C;gBACI,sBAAO,O;gBAAP, wB;;;YAGR,sBAAO,E;;;U+KrHH,iD;UAGI,gCAA2B,EAA3B,C;YAHJ,2BAGqC,I;IBACjC,IAAK,a3LyxD0C,I2L zxD1C,gBAAyB,uBAAzB,CAAL,C;YAJJ,2B3L6xDmD,IOjmDsB,WoLxLI,0BAAuC,mBAAvC,IpLwLJ,C;;YoL5 LzE,2BAKY,I;;UAYeR,iE9LJD,yB8LIC,4B3L+sD+C,I;;QA1RpB,8B;UAA6C,6B;;;M2LpgDhF,OAIkFk,S3Lo7CE, W2Lp7CF,EAAO,mBAAC,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAvET,+B;MAeyC,gCAAC,EAAd,C;K;IA EzC,6C;MAGgC,yB;QAAA,YAAoB,E;MAM3C,Q;MALL,cAAY,gB;M3LurBL,kBAAS,gB;MA2FA,U;MAAA,S2 LhxBM,O3LgxBN,W;MAAhB,OAAGb,gBAAhB,C;QAAGb,2B;QAAM,Ia3hB6B,CAAC,Qb2hBhB,Oa3hBgB,Cb 2hB9B,C;UAAwB,WAAy,WAAI,OAAJ,C;;M2L9wBrD,kB3L+wBE,W;MAMrBA,oBAAM,iBAAa,qCAAwB,EA AxB,CAAb,C;MAuEA,U;MAAA,+B;MAAb,OAAa,gBAAb,C;QAAA,wB;QACT,aAAY,uBAAC,IAAd,E;;M2L5gD hB,sBAAsB,CAGjB,oB3L0gDE,a2L1gDF,CAHiB,mBAGF,C;MAEP,yBAAS,mBAAS,YAAA,SAAU,OAAV,EA AmB,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAAkB,SAAlB,C;MAMc9D,gBAAGb,iBAnCT,OAmCS,C;M3 Lg7CT,oBAAoB,gB;MAoSd,kB;MADb,YAAY,C;MACC,S2LvDN,O3LuvDM,W;MAAb,OAAa,gBAAb,C;QAA a,0B;QA1RsB,U;QAAA,cA0RT,oBAAMb,cAAnB,EAAMb,sBAAnB,U;Q2L/sDIB,kB;Q3Lq7C2B,c2Lx7C3B,CA AC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,M2LltDH,C3Lw7CjB,G2Lv7C3B,I3Lu7C2B,G2Lr7C3B,oBA xCmG,Q3LuvDpD,M2LvDoD,kBAwCnG,Y9LJD,yB8LIC,4B3L+sD+C,MA1RpB,U;UAA6C,+B;;;M2L79ChF,O A0CK,S3Lo7CE,a2Lp7CF,EAAO,mBAAC,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAjCI,8C;MAAA,qB;QAE G,IAAG,QAAG,EAAG,CAAG,C;UAEQ,IAAA,EAAG,OAAG,GAAY,cAAO,OAAnB,C;YAHZ,OAGyC,c;;YAHZ C,OAIoB,E;;UAJpB,OAoy,iBAAS,E;O;K;IAfjC,0C;MAKgC,sB;QAAA,SAAiB,M;MAC7C,OAYK,eAXA,OAD L,uBACK,EAAI,4BAAJ,CAWA,EAAa,IAAb,C;K;IAET,gC;MAAwC,uB;;Q/KmDtB,gC;QAAA,gC;QAAA,mB;Q AAA,kB;QAAA,kB;QAAd,0D;UACI,I+KpD+C,CAAI,aAAH,U/KoDIC,iCAAK,KAAL,E+KpDkC,CAAG,C/KoD nD,C;YACI,sBAAO,K;YAAP,wB;;;QAGR,sBAAO,E;;;Mf3CA,4B;M8Lb6B,OAA8C,OAAM,EAAV,GAAC,gBA Ad,GAA0B,E;K;IAGpF,wC;MAAkB,W;K;IAC9B,oD;MAAA,uB;QAAkB,wBAAS,I;O;K;IAFvC,mC;MACI,IAA A,M9KkMgD,YAAU,C8KIM1D,C;QAD4C,OACxB,wB;;QADwB,OAEPc,kC;K;mBAGZ,yB;M3L86CA,+D;MA oSA,wE;M2LltDA,sF;QAKI,gBAAGb,2B;Q3Lg7CT,kBAAoB,gB;QAoSd,gB;QADb,YAAY,C;QACC,2B;QAAb,

OAAa,cAAb,C;UAAa,sB;UA1RsB,U;UAAA,cA0RT,oBAAmB,cAAAnB,EAAmB,sBAAnB,U;U2L/sDIB,kB;U3Lq  
7C2B,c2Lx7C3B,CAAC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,I2LltDH,C3Lw7CjB,G2Lv7C3B,I3Lu7C  
2B,G2Lr7C3B,sC3L+sD+C,I2L/sD/C,a9LJD,yB8LIC,4B3L+sD+C,IA1RpB,U;YAA6C,6B;;Q2Lz7ChF,OAMK,S3  
Lo7CE,W2Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;O;KAbT,C;6EvEkSA,0B;MAGmE,O  
AAA,SAAK,gBAAO,GAAP,C;K;qFAExE,yB;MAAA,yD;MAAA,gC;QAO2B,gBAAhB,oB;QAAsB,avHrU7B,W;  
QuHqUA,OvHpUO,SuHoUqC,W;O;KAPhD,C;uFAUA,yB;MAAA,iE;MAAA,0C;QAQmC,gBAAXB,mBAAc,QA  
Ad,C;QAA8B,avHhVrC,W;QuHgVA,OvH/uo,SuH+U6C,W;O;KARxD,C;IAWA,oC;MAliB,Q;MAAb,wBAAa,K  
AAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;IAGX,oC;MAliB,Q;MAAb,wBAAa  
,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;qFAGX,qB;MAG8D,gCAAO,EA  
AP,C;K;qFAE9D,4B;MAGkF,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAO9D,4B;MAG4E,OAAA,y  
BAAO,KAAP,CAVd,gBAAO,EAAP,C;K;qFAY9D,4B;MAGyE,OAAA,yBAAO,KAAP,CafX,gBAAO,EAAP,C;  
K;qFAiB9D,4B;MAG8E,OAAA,yBAAO,KAAP,CAPhB,gBAAO,EAAP,C;K;qFAsB9D,4B;MAGyE,OAAA,yB  
AAO,KAAP,CAzBX,gBAAO,EAAP,C;K;qFA2B9D,4B;MAG4E,OAAA,yBAAO,KAAP,CA9Bd,gBAAO,EAAP,  
C;K;I/H/a9D,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MAQmB,Q;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;M  
AAL,iB;QAA2B,OAAO,I;;MAA5C,UAAU,I;MACV,IAAI,MAAM,sCAAK,UAAW,IAAwB,MAAM,sCAAK,UA  
AvC,C;QAAkD,OAAO,I;MACzD,OAAW,OAAJ,GAAL,C;K;IAGf,kC;MAK4C,kCAAsB,EAAtB,C;K;IAE5C,2C;  
MAQmB,Q;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;;MAA5C,UAAU,I;MACV,IAAI  
,MAAM,uCAAM,UAAZ,IAAyB,MAAM,uCAAM,UAAzC,C;QAAoD,OAAO,I;MAC3D,OAAW,QAAJ,GAAL,C;  
K;IAGf,gC;MAKwC,gCAAoB,EAAPb,C;K;IAExC,yC;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,  
IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAGb,qBAAK,CAAL,C;MACH  
B,IAAI,YAAY,EAAb,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAEExB,QAAQ,C;QAER,IAAI,cAAa,E  
AAjB,C;UACI,aAAa,I;UACb,QAAQ,W;eACL,IAAI,cAAa,EAajB,C;UACH,aAAa,K;UACb,QAAQ,W;;UAER,O  
AAO,I;;QAEEX,QAAQ,C;QACR,aAAa,K;QACb,QAAQ,W;;MAIZ,uBAAuB,S;MAEvB,qBAAqB,gB;MACrB,aAA  
a,C;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAaiB,KAAjB,C;QA  
EZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACTB,IAAI,SAAS,cAAb,C;UACI,IAAI,mBAAkB,gBAAtB,C;YACI,i  
BAAiB,QAAQ,KAAR,I;YAEjB,IAAI,SAAS,cAAb,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAlf,6BAAU,KAAV,C;  
QAEA,IAAI,UAAAS,QAAQ,KAAR,IAAT,CAAJ,C;UAA4B,OAAO,I;QAEEnC,kBAAU,KAAV,I;;MAGJ,OAAW,U  
AAJ,GAAgB,MAAhB,GAA4B,CAAC,MAAD,I;K;IAGvC,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MAQI,W  
AAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MA  
CA,S;MAEA,gBAAGb,qBAAK,CAAL,C;MACHB,IAAI,YAAY,EAAb,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,  
OAAO,I;QAEExB,QAAQ,C;QAER,IAAI,cAAa,EAajB,C;UACI,aAAa,I;UACb,gC;eACG,IAAI,cAAa,EAajB,C;U  
ACH,aAAa,K;UACb,6B;;UAEA,OAAO,I;;QAEEX,QAAQ,C;QACR,aAAa,K;QACb,6B;;MAIJ,2C;MAEA,qBAAq  
B,gB;MACrB,e;MACA,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAaiB,K  
AAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACTB,IAAI,uBAAS,cAAT,KAAJ,C;UACI,IAAI,uBAAK  
B,gBAAIB,CAAJ,C;YACI,iBAAiB,8BAAQ,KAAR,E;YAEjB,IAAI,uBAAS,cAAT,KAAJ,C;cACI,OAAO,I;;YAG  
X,OAAO,I;;QAlf,6CAAU,KAAV,E;QAEA,IAAI,uBAAS,8BAAQ,KAAR,EAAT,KAAJ,C;UAA4B,OAAO,I;QAE  
nC,6CAAU,KAAV,E;;MAGJ,OAAW,UAAJ,GAAgB,MAAhB,GAA6B,MAAD,a;K;IAIvC,kC;MAAyD,MAAM,0  
BAAsB,6BAA0B,KAA1B,MAAtB,C;K;uEwBhI/D,yB;MAAA,oC;MAAA,uC;QAIi,iBAAiB,C;QACjB,eAAe,mB  
AAS,CAAT,I;QACf,iBAAiB,K;QAEjB,OAAO,cAAc,QAARb,C;UACI,YAAGb,CAAC,UAAL,GAAiB,UAAjB,G  
AAiC,Q;UAC7C,YAAY,UAAU,iCAAk,KAAL,EAAY,C;UAEZ,IAAI,CAAC,UAAL,C;YACI,IAAI,CAAC,KAA  
L,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cAEA,sBAAY,CAAZ,I;;QAIZ,O  
AAO,8BAAY,UAAZ,EAAb,WAAW,CAAX,IAAxB,C;O;KAZBX,C;yEA4BA,yB;MAAA,8B;MA5BA,oC;MA4  
BA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QA5BD,iBAAiB,C;QACjB,eAAe,qBAAS,CAAT,I;QACf,iBAAiB,K;QAEj  
B,OAAO,cAAc,QAARb,C;UACI,YAAGb,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;UAC7C,YAsBwB,SAtBZ,CA  
AU,mCAAK,KAAL,EAAY,C;UAEZ,IAAI,CAAC,UAAL,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;;cAEb,0B  
AAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cAEA,sBAAY,CAAZ,I;;QAWZ,OAPO,gCAAY,UAAZ,EA  
AwB,WAAW,CAAX,IAAxB,CAOGC,W;O;KAJ3C,C;iFAMA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIuB,UA  
AL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI



,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QAEf,OAAO,E;O;KARX,C;mFAWA,yB;MAAA,8B;MAXA,mD;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAAsB,oB;;UAXJ,kC;UAAA,qBAAL,WAAK,C;UAAL,qB;UAAA,oB;UAAA,oB;UAAAd,0D;YACI,IAAI,CAUyB,SAVxB,CAAU,mCAAK,KAAL,EAAV,CAAL,C;cACI,mBAAO,gCAAY,KAAZ,EAAmB,kBAAnB,C;cAAP,qB;;UAER,mBAAO,E;;;QAOP,OAA4C,2B;O;KAJhD,C;6EAMA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;;QAEf,OAAO,E;O;KARX,C;+EAWA,yB;MAAA,8B;MAXA,mD;MAAA,+C;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,kB;;UAXT,U;UAAA,SAAa,SAAR,YAAL,WAAK,CAAQ,CAAb,W;UAAAd,OAAc,gBAAd,C;YAAc,yB;YACV,IAAI,CAUuB,SAVtB,CAAU,mCAAK,KAAL,EAAV,CAAL,C;cACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;cAAP,mB;;UAER,iBAAO,E;;;QAOP,OAA0C,yB;O;KAJ9C,C;IAMA,kC;MAhEI,iBAAiB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAAR,B,C;QACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YA6DgE,4BA7D1C,iCAAK,KAAL,EA6D0C,E;QA3DhE,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAkDiD,OA9CtD,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;IAGDX,kC;MAzCK,Q;MAAsB,kBAAtB,2D;MA5BD,iBAAiB,C;MACjB,eAAe,qBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAAR,B,C;QACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YAkEoD,4BAIE9B,mCAAK,KAAL,EAKe8B,E;QAhEpD,IAAI,CAAC,UAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAuDqC,OAnD1C,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOGC,W;K;IA8C3C,uC;MAGsE,oB;;QA3C/C,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CA0CsE,4BA1C3D,iCAAK,KAAL,EA0C2D,EA1C1E,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;;MAuC2D,uB;K;IAEtE,uC;MAICK,Q;MAAsB,kBAAtB,2D;MAAsB,oB;;QAXJ,kC;QAAA,wBAAL,WAAK,C;QAAL,qB;QAAA,oB;QAAA,oB;QAAd,0D;UACI,IAAI,CA+C0D,4BA/C/C,mCAAK,KAAL,EA+C+C,EA/C9D,C;YACI,mBAAO,gCAAY,KAAZ,EAAmB,kBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;;MA4C+C,OArCV,2B;K;IAuChD,qC;MAGoE,kB;;QApCID,Q;QAAA,OAaA,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,CAmCkE,4BAnCvD,iCAAK,KAAL,EAmCuD,EAnCtE,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;;MAGCyD,qB;K;IAEpE,qC;MA3BK,Q;MAAsB,kBAAtB,2D;MAAsB,kB;;QAXT,U;QAAA,SAAa,WAAR,eAAL,WAAK,CAAQ,CAAb,W;QAAd,OAAc,gBAAd,C;UAAc,yB;UACV,IAAI,CAwCsD,4BAxC3C,mCAAK,KAAL,EAwC2C,EAxC1D,C;YACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;;MAqC6C,OA9BV,yB;K;IAGC9C,2B;MA9FI,iBAAiB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAAR,B,C;QACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,mCAAsB,iCAAK,KAAL,EAAtB,E;QAEA,IAAI,CAAC,UAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAGf+B,OA5EpC,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;yEA8EX,yB;MAAA,8B;MAAA,qC;MAAA,4B;QAI2C,Q;QAAD,OAAuB,KAAtB,2DAAsB,CAAO,W;O;KAJxE,C;IAMA,gC;MAGoD,oB;;QAI1E7B,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAAJ,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;;MAsEyC,uB;K;mFAEpD,yB;MAAA,8B;MAAA,+C;MAAA,4B;QAIgD,Q;QAAD,OAAuB,UAAtB,2DAAsB,CAAY,W;O;KAJIF,C;IAMA,8B;MAGkD,kB;;QApEhC,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAAJ,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;;MAGEuC,qB;K;+EAEID,yB;MAAA,8B;MAAA,2C;MAAA,4B;QAI8C,Q;QAAD,OAAuB,QAAtB,2DAAsB,CAAU,W;O;KAJ9E,C;IAMA,8C;MAU8C,uB;QAAA,UAAgB,E;MAO5C,Q;MANd,IAAI,SAAS,CAAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAAY,mBAAL,SAAK,EAAAY,CAAZ,EAAe,SAAK,OAApB,C;MAEhB,SAAS,mBAAc,MAAd,C;MACK,gBAAS,SAAK,OAAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAAG,gBAAO,OAAP,C;MACP,EAAG,gBAAO,SAAP,C;MACH,OAAO,E;K;IAGX,gD;MASwC,uB;QAAA,UAAgB,E;MACnD,Q;MAAD,OAAuB,SAAtB,6DAAsB,EAAS,MAAT,EAAiB,OAAjB,CAA0B,W;K;IAErD,4C;MAU4C,uB;QAAA,UAAgB,E;MAQ1C,Q;MAPd,IAAI,SAAS,CAAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB

,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,SAAK,OAApB,C; MAEhB,SAAS,mBAAC,MAAd,C;MACT,EAAG,gBAAO,SAAP,C;MACW,gBAAS,SAAK,OAAAd,I;MAAd,aAAU ,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,OAAO,E;K;IAGX,8C;MASsC,uB;QAAA,UAAgB,E;MACjD, Q;MAAD,OAAuB,OAAtB,6DAAsB,EAAO,MAAP,EAAe,OAaf,CAAwB,W;K;2FAEnD,qB;MAWI,OAAO,qBA AgB,SAAK,OAAL,KAAe,C;K;+EAG1C,qB;MAMoD,4BAAU,C;K;sFAE9D,qB;MAMuD,0BAAS,C;K;mFAMhE, yB;MAAA,2C;MAAA,4B;QAMuD,QAAC,kB;O;KANxD,C;yFAQA,yB;MAAA,2C;MAAA,4B;QAWI,OAAO,qB AAqB,QAAL,SAAK,C;O;KAXhC,C;IAiB4D,+C;MAAA,kC;MAAS,uB;MACjE,eAAoB,C;K;gDAEpB,Y;MAA2C ,gB;MAAA,iE;MAAJ,4C;K;+CAEvC,Y;MAAyC,sBAAQ,yB;K;;IARrD,+B;MAG4D,4C;K;+EAQ5D,qB;MAE8C, uCAAQ,E;K;+EAETd,mC;MASI,OA5DgD,qBAAU,CA4D1D,GAAe,cAAf,GAAmC,S;K;6EAEvC,yB;MAAA,2C; MAAA,0C;QASI,OAAI,kBAAJ,GAAe,cAAf,GAAmC,S;O;KATvC,C;IAeI,mC;MAAQ,uBAAg,mBAAS,CAAT,I AAH,C;K;IAMR,qC;MAAQ,OAAA,SAAK,OAAL,GAAc,CAAd,I;K;IAEZ,8C;MAIuB,Q;MAAA,0BAAS,CAAT, I;MAAnB,OAAGB,CAAT,8BACgB,gBAAZ,qBAAK,KAAL,CAAY,CADhB,IAEoB,eAAhB,qBAAK,QAAQ,CA AR,IAAL,CAAgB,C;K;IAG/B,uC;MAGuD,ONpKyC,oBMoK/B,KAAM,MNpKyB,EMoKIB,KAAM,aAN,GAA qB,CAArB,INpKkB,C;K;IMsKhG,yC;MAGqE,qCAAY,KAAM,MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,I AAzB,C;K;uFAErE,iC;MAS2E,2BAAy,KAAZ,EAAMb,GAAAnB,C;K;mFAE3E,2C;MAO0D,wB;QAAA,WAAgB ,gB;MAAkB,OAAA,8BAAy,UAAZ,EAAwB,QAAxB,CAAKC,W;K;IAE9H,uC;MAG6D,OAAA,8BAAy,KAAM, MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,IAAzB,CAAiD,W;K;IAE9G,sE;MAImD,qC;QAAA,wBAAgC,S; MAC/E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GN1M4F,oBM0M/B,CN1M+ B,EM0M5B,KN1M4B,C;K;IM6MhG,wE;MAIqD,qC;QAAA,wBAAgC,S;MACjF,YAAy,sBAAQ,SAAR,C;MAC Z,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GNnN4F,oBMmN/B,CNnN+B,EMmN5B,KNnN4B,C;K;IMsNhG,qE ;MAIkD,qC;QAAA,wBAAgC,S;MAC9E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBA AxB,GN5N4F,oBM4N/B,QAAQ,CAAR,IN5N+B,EM4NpB,gBN5NoB,C;K;IM+NhG,uE;MAIoD,qC;QAAA,wBA AgC,S;MACHf,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GNrO4F,oBMqO/B, QAAQ,SAAU,OAAIB,INrO+B,EMqOL,gBNrOK,C;K;IMwOhG,0E;MAIuD,qC;QAAA,wBAAgC,S;MACnF,YA AY,0BAAy,SAAZ,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GN9O4F,oBM8O/B,CN9O+B,EM8O5B, KN9O4B,C;K;IMiPhG,4E;MAIyD,qC;QAAA,wBAAgC,S;MACrF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAA S,EAAPB,GAAwB,qBAAxB,GNvP4F,oBMuP/B,CNvP+B,EMuP5B,KNvP4B,C;K;IM0PhG,yE;MAIsD,qC;QAAA ,wBAAgC,S;MACIF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GNhQ4F,oBMg Q/B,QAAQ,CAAR,INhQ+B,EMgQpB,gBNhQoB,C;K;IMmQhG,2E;MAIwD,qC;QAAA,wBAAgC,S;MACpF,YA AY,0BAAy,SAAZ,C;MACZ,OAAW,UAAAS,EAAPB,GAAwB,qBAAxB,GNzQ4F,oBMMyQ/B,QAAQ,SAAU,OAA IB,INzQ+B,EMyQL,gBNzQK,C;K;IM4QhG,oE;MAOI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,Q AAAb,oCAAKD,UAAID,OAA1B,C;MACV,SAAS,sB;MACT,EAAG,qBAAy,SAAZ,EAakB,CAAIB,EAAqB,UAA rB,C;MACH,EAAG,gBAAO,WAAP,C;MACH,EAAG,qBAAy,SAAZ,EAakB,QAAlB,EAA4B,gBAA5B,C;MAC H,OAAO,E;K;yFAGX,yB;MAAA,8B;MAAA,qD;MAAA,+D;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,U AAAb,EAAYB,QAazB,EAAmC,WAAAnC,CAAGD,W;O;KAP3E,C;IASA,uD;MAOI,+BAAa,KAAM,MAAnB,EAA 0B,KAAM,aAN,GAAqB,CAArB,IAA1B,EAakD,WAAID,C;K;yFAEJ,yB;MAAA,8B;MAAA,qD;MAAA,gD;Q AOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,KAAb,EAAoB,WAApB,CAAiC,W;O;KAP5D,C;IASA,sD;MASI ,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAKD,UAAID,OAA1B,C;MAEV,IAAI,aAAY, UAAhB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,gBAAf,C;MAEhB,SAAS,mBAAC,oBAAU,QAAY, GAAqB,UAArB,KAAd,C;MACT,EAAG,qBAAy,SAAZ,EAakB,CAAIB,EAAqB,UAArB,C;MACH,EAAG,qBA AY,SAAZ,EAakB,QAAlB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;uFAGX,yB;MAAA,8B;MAAA,mD;MAAA,k D;QASK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,UAAZ,EAAwB,QAAxB,CAAKC,W;O;KAT7D,C;IAWA,yC ;MAKqE,8BAAy,KAAM,MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,IAAzB,C;K;uFAErE,yB;MAAA,8B;M AAA,mD;MAAA,mC;QAOK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,KAAZ,CAAmB,W;O;KAP9C,C;IASA, yC;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,OAAO,8BAAy,MAAO,OAAAnB,EAA2B,gBAA3B,C;;MAEX,O AAO,8BAAy,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,ON3XyE,oBM2 XxD,MAAO,ON3XiD,C;;MM6X7E,OAAO,S;K;IAGX,yC;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,OAAO,8B AAY,CAAZ,EAAe,mBAAS,MAAO,OAAhB,IAAf,C;;MAEX,OAAO,8BAAy,CAAZ,EAAe,gBAAf,C;K;IAGX,2

C;MAKI,IAAI,sBAAS,MAAT,CAAI,C;QACI,ON9YwF,oBM8YvE,CN9YuE,EM8YpE,mBAAS,MAAO,OAAhB,IN9YoE,C;;MMgz5F,OAAO,S;K;IAGX,sD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAGB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAmE,sBAAS,MAAT,CAAvE,C;QACI,OAAO,8BAAY,MAAO,OAAnB,EAA2B,mBAAS,MAAO,OAAhB,IAA3B,C;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,wD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAGB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAmE,sBAAS,MAAT,CAAvE,C;QACI,ONTawF,oBMsavE,MAAO,ONtagE,EMsaxD,mBAAS,MAAO,OAAhB,INtawD,C;;MMwa5F,OAAO,S;K;IAGX,mD;MAKmF,oCAAkB,SAAIB,EAA6B,SAA7B,C;K;IAEnF,mD;MAKuE,sCAAkB,SAAIB,EAA6B,SAA7B,C;K;IAEvE,iF;MAIsE,qC;QAAA,wBAAGC,S;MACIG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QA5JvB,U;QA4JM,OA5JgB,aAAtB,+DAAsB,EA4JyC,CA5JzC,EA4J4C,KA5J5C,EA4JmD,WA5JnD,CAAGD,W;;MA4JvE,W;K;IAGJ,mF;MAIwE,qC;QAAA,wBAAGC,S;MACpG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QArKvB,U;QAqKM,OArKgB,aAAtB,+DAAsB,EAqKyC,CArKzC,EAqK4C,KArK5C,EAqKmD,WArKnD,CAAGD,W;;MAqKvE,W;K;IAGJ,gF;MAIqE,qC;QAAA,wBAAGC,S;MACjG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QA9K1E,U;QA8KM,OA9KgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EA8K4D,WA9K5D,CAAGD,W;;MA8KvE,W;K;IAGJ,kF;MAIuE,qC;QAAA,wBAAGC,S;MACnG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAvLzF,U;QAuLM,OAvLgB,aAAtB,+DAAsB,EAaA,UAAb,EAAYB,QAazB,EAuL2E,WAvL3E,CAAGD,W;;MAuLvE,W;K;IAGJ,oF;MAI2E,qC;QAAA,wBAAGC,S;MACvG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAhMzF,U;QAgMM,OAHMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAgM2E,WAhM3E,CAAGD,W;;MAGMvE,W;K;IAGJ,sF;MAIyE,qC;QAAA,wBAAGC,S;MACrG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QAzM1E,U;QAYMM,OAzMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAyM4D,WAzM5D,CAAGD,W;;MAYMvE,W;K;IAGJ,qF;MAI0E,qC;QAAA,wBAAGC,S;MACtG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAINvB,U;QAKNM,OAINgB,aAAtB,+DAAsB,EAKNyC,CAINzC,EAKN4C,KAIN5C,EAKNmD,WAINnD,CAAGD,W;;MAKNvE,W;K;IAGJ,uF;MAI4E,qC;QAAA,wBAAGC,S;MACxG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QA3NvB,U;QA2NM,OA3NgB,aAAtB,+DAAsB,EA2NyC,CA3NzC,EA2N4C,KA3N5C,EA2NmD,WA3NnD,CAAGD,W;;MA2NvE,W;K;+EAOJ,yC;MAQoF,OAAA,KAAM,iBAAQ,SAAR,EAAC,WAAd,C;K;+EAE1F,uC;MAOI,OAAA,KAAM,iBAAQ,SAAR,EAAC,SAAd,C;K;yFAEV,yC;MAMYf,OAAA,KAAM,sBAAa,SAAb,EAAmB,WAAAnB,C;K;+FAE/F,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAEw,Q;QAAA,IApe4C,mBAAS,CAoerD,C;uBAAkB,oBAAU,iCAAK,CAAL,EAAV,E;UAAA,YNljBoD,oBMkjBrB,CNljBqB,C;UMkjBtE,OLrjBwD,2BAAL,GAakB,K;;UKqjBrE,OAAYD,S;QAaHE,W;O;KafJ,C;iGakBA,yB;MAAA,oC;MAAA,uC;QAEI,OAtfmD,mBAAS,CAsf5D,GAAYB,UAAU,iCAAK,CAAL,EAAV,CAAmB,WAAAnB,GNpkBoD,oBMokBV,CNpkBU,CMokB7E,GAA2E,S;O;Kaf/E,C;+EAmBA,4B;MAIsE,OAAA,KAAM,iBAAQ,SAAR,C;K;IAE5E,0F;MAKI,IAAK,cAAc,CAAf,IAAsB,aAAa,CAAnC,IAA0C,cAAa,SAAK,OAAL,GAAC,MAAd,IAAb,CAA1C,IAAiF,eAAc,KAAM,OAAN,GAae,MAAf,IAAd,CAArF,C;QACI,OAAO,K;;MAGX,iBAAC,CAAd,UAAsB,MAAtB,U;QACI,IAAI,CAA0B,SAAZB,qBAAK,aAAa,KAAb,IAAL,CAAYB,EAAO,iBAAM,cAAc,KAAd,IAAN,CAAP,EAAmC,UAAAnC,CAA9B,C;UACI,OAAO,K;;MAEf,OAAO,I;K;IAGX,mD;MAG+C,0B;QAAA,aAAsB,K;MACjE,OAAA,SAAK,OAAL,GAAC,CAAd,IAA2B,SAAR,qBAAK,CAAL,CAAQ,EAAO,IAAP,EAAa,UAAb,C;K;IAE/B,iD;MAG6C,0B;QAAA,aAAsB,K;MAC/D,OAAA,SAAK,OAAL,GAAC,CAAd,IAAmC,SAAhB,qBAAK,2BAAL,CAAGB,EAAO,IAAP,EAAa,UAAb,C;K;IAEvC,qD;MAGyD,0B;QAAA,aAAsB,K;MAC3E,IAAI,CAAC,UAAD,IAAE,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,WAAL,SAAK,EAAW,MAAX,C;;QAEZ,OAAO,6BAakB,CAaIB,EAAqB,MAArB,EAA6B,CAA7B,EAAGC,MAAO,OAAvC,EAA+C,UAA/C,C;K;IAGf,iE;MAG0E,0B;QAAA,aAAsB,K;MAC5F,IAAI,CAAC,UAAD,IAAE,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,aAAL,SAAK,EAAW,MAAX,EAAmB,UAAAnB,C;;QAEZ,OAAO,6BAakB,UAAIB,EAA8B,MAA9B,EAAc,CAAtC,EAAYC,MAAO,OAAhD,EAawD,UAAxD,C;K;IAGf,mD;MAGuD,0B;QAAA,aAAsB,K;MACzE,IAAI,CAAC,UAAD,IAAE,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,SAAL,SAAK,EAAS,MAAT,C;;QAEZ,OAAO,6BAakB,mBAAS,MAAO,OAAhB,IAAIB,EAA0C,MAA1C,EAakD,CAAID,EAAqD,M

AAO,OAA5D,EAAoE,UAApE,C;K;IAMf,wD;MAQ8D,0B;QAAA,aAAsB,K;MACHf,qBfjnBO,MAAO,KeinBa,S  
AAK,OfjnBIB,EeinB0B,KAAM,OfjnBhC,C;MemnBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA8B,SAAR,qBAA  
K,CAAL,CAAQ,EAAO,iBAAM,CAAN,CAAP,EAA8B,UAA9B,CAArC,C;QACI,a;MAEJ,IAAS,mBAAL,SAAK  
,EAAmB,IAAI,CAAJ,IAAnB,CAAL,IAAwC,mBAAN,KAAM,EAAmB,IAAI,CAAJ,IAAnB,CAA5C,C;QACI,a;;  
MAEJ,OAAO,8BAAY,CAAZ,EAAe,CAAf,CAAkB,W;K;IAG7B,wD;MAQ8D,0B;QAAA,aAAsB,K;MACHf,iBA  
AiB,SAAK,O;MACtB,kBAAkB,KAAM,O;MACxB,qBfxoBO,MAAO,KewoBa,UfxoBb,EewoByB,WfxoBzB,C;M  
e0oBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA+C,SAAzB,qBAAK,aAAa,CAAb,GAAiB,CAAjB,IAAL,CAAyB,  
EAAO,iBAAM,cAAc,CAAd,GAakB,CAAIB,IAAN,CAAP,EAAgD,UAAhD,CAAtD,C;QACI,a;MAEJ,IAAS,mB  
AAL,SAAK,EAAmB,aAAa,CAAb,GAAiB,CAAjB,IAAnB,CAAL,IAAqD,mBAAN,KAAM,EAAmB,cAAc,CAAd  
,GAakB,CAAIB,IAAnB,CAAZD,C;QACI,a;MAEJ,OAAO,8BAAY,aAAa,CAAb,IAAZ,EAA4B,UAA5B,CAAwC  
,W;K;IAMnD,8D;MAQqD,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAMnE,UAAkB,M;MAL3C,IAAI,  
CAAC,UAAD,IAAe,KAAM,OAAN,KAAC,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB,  
ONjtBwF,kB8G3ME,oBxG45BrE,IwG55BqE,C9G2MF,EMitB7D,UNjtB6D,C;;MMotBnE,uBAAX,UAAW,EAAC  
,CAAd,C;MAAkB,oC;kBAA3C,gD;QACI,kBAAkB,qBAAI,KAJ,C;QACR,c;;UjCikXE,U;UAAhB,4BiCjkXQ,Kj  
CikXR,kB;YAAgB,cAAhB,UiCjkXQ,KjCikXR,S;YAAAsB,IiCjkXC,SAAH,UjCikXgB,oBiCjkXhB,CAAG,0BjCik  
XD,C;cAAwB,aAAO,I;cAAP,e;;;UAC9C,aAAO,K;;;QiClkXH,e;UACI,OAAO,K;;MAEf,OAAO,E;K;IAGX,KE;M  
ASyD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACxG,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,K  
AAc,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONruB4F,sB8G3MM,oBxGg7BzE,IwG  
h7ByE,C9G2MN,EMquB7D,UNruB6D,C;;kBMyuBhG,iBAAYB,eAAX,UAAW,EAAa,2BAAb,CAAzB,WAAwD,  
CAAxD,U;QACI,kBAAkB,qBAAI,KAJ,C;QACR,c;;UjCyiXE,Q;UAAhB,wBiCziXQ,KjCyiXR,gB;YAAgB,cAA  
hB,UiCziXQ,KjCyiXR,O;YAAAsB,IiCziXC,SAAH,UjCyiXgB,oBiCziXhB,CAAG,0BjCyiXD,C;cAAwB,aAAO,I;c  
AAP,e;;;UAC9C,aAAO,K;;;QiC1iXH,e;UACI,OAAO,K;;MAGf,OAAO,E;K;IAIX,8E;MAA2G,oB;QAAA,OAAG  
B,K;MAOrG,UAKA,M;MAXIB,cAAkB,CAAC,IAAL,GACV,aAAW,gBAAX,UAAW,EAAC,CAAd,CAAX,EAA  
S,C,eAAT,QAAS,EAAa,gBAAb,CAAtC,CADU,GAGV,SAAW,eAAX,UAAW,EAAa,2BAAb,CAAX,EAAmD,gBA  
AT,QAAS,EAAC,CAAd,CAAnD,C;MAEJ,IAAI,iCAAkB,yBAAtB,C;QACkB,yB;QAAd,OAAC,cAAd,C;UAAc,u  
B;UACV,IAAU,cAAN,KAAM,EAAC,CAAd,EAAiB,SAAJB,EAAuB,KAAvB,EAA8B,KAAM,OAAPC,EAA4C,U  
AA5C,CAAV,C;YACI,OAAO,K;;;QAGD,2B;QAAd,OAAC,gBAAd,C;UAAc,2B;UACV,IAAU,kBAAN,KAAM,E  
AAkB,CAAIB,EAAqB,SAArB,EAA2B,OAA3B,EAAkC,KAAM,OAAXC,EAAgD,UAAhD,CAAV,C;YACI,OAA  
O,O;;;MAGnB,OAAO,E;K;IAGX,qE;MAUsB,UAMA,M;MAfIB,IAAI,CAAC,UAAD,IAAe,OAAQ,KAAR,KAAG  
B,CAAN,C;QACI,aAAqB,UAR,OAAQ,C;QACrB,YAAgB,CAAC,IAAL,GAAW,sBAAQ,MAAR,EAAgB,UA  
AhB,CAAX,GAA4C,0BAAY,MAAZ,EAAoB,UAApB,C;QACxD,OAAW,QAAQ,CAAZ,GAAe,IAAf,GAAyB,U  
AAS,MAAT,C;;MAGpC,cAAkB,CAAC,IAAL,GAAW,aAAW,gBAAX,UAAW,EAAC,CAAd,CAAX,EAA6B,gB  
AA7B,CAAX,GAAoD,SAAW,eAAX,UAAW,EAAa,2BAAb,CAAX,EAA0C,CAA1C,C;MAEIE,IAAI,6BAAJ,C;Q  
ACKB,yB;oBAAd,OAAC,cAAd,C;UAAc,yB;UACmB,sB;;Yb7sBrB,U;YAAA,Sa6sBa,Ob7sBb,W;YAAhB,OAAG  
B,gBAAhB,C;cAAGB,2B;cAAM,Ia6sBgC,cb7sBIB,Oa6sBkB,EAAC,CAAd,sBb7sBIB,Oa6sBmD,OAAjC,ab7sBhC  
,C;gBAAwB,qBAAO,O;gBAAP,uB;;;YAC9C,qBAAO,I;;;Ua4sBC,uC;UACA,IAAI,sBAAJ,C;YACI,OAAO,YAA  
S,cAAT,C;;;QAGD,2B;oBAAd,OAAC,gBAAd,C;UAAc,2B;UACmB,wB;;YbntBrB,U;YAAA,SamtBa,ObntBb,W;  
YAAhB,OAAGB,gBAAhB,C;cAAGB,6B;cAAM,IamtBgC,kBbntBIB,SamtBkB,EAAkB,CAAIB,sBbntBIB,SamtBu  
D,OAARc,abntBhC,C;gBAAwB,uBAAO,S;gBAAP,uB;;;YAC9C,uBAAO,I;;;UaktBC,2C;UACA,IAAI,wBAAJ,C;  
YACI,OAAO,YAAS,gBAAT,C;;;MAInB,OAAO,I;K;IAGX,iE;MAY+D,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,  
aAAsB,K;MACtG,4BAAU,OAAV,EAAmB,UAnB,EAA+B,UAA/B,EAakD,KAAID,C;K;IAEJ,mE;MAYmE,0B  
;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACIH,4BAAU,OAAV,EAAmB,UAnB,EAA+B,UAA/B,EA  
AkD,IAAID,C;K;IAEJ,KE;MAWgE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvG,gB;MAAA,8CA  
AU,OAAV,EAAmB,UAnB,EAA+B,UAA/B,EAakD,KAAID,mDAAmE,E;K;IAEvE,sE;MAYoE,0B;QAAA,aA  
AkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnH,gB;MAAA,8CAAU,OAAV,EAAmB,UAnB,EAA+B,UAA/B,EA  
AkD,IAAID,mDAaKE,E;K;IAKtE,6D;MAM4C,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACnF,OAA  
W,cAAc,gCAAzB,GACI,sBAAW,mBAAY,IAAZ,CAAX,EAA8B,UAA9B,EAA0C,UAA1C,CADJ,GNz2B4F,kB8  
G3ME,oBxGujC5E,IwGvjC4E,C9G2MF,EM42BpE,UN52BoE,C;K;IM+2BhG,+D;MAQgD,0B;QAAA,aAAkB,C;

MAAG,0B;QAAA,aAAsB,K;MACvF,OAAW,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAgB,UAAhB,EAA4B,gB  
AA5B,EAAoC,UAApC,CADJ,GNx3B4F,kBM23B1E,MN33B0E,EM23BIE,UN33BkE,C;K;IM83BhG,iE;MAQgD  
,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MAC/F,OAAW,cAAc,gCAAzB,GACI,0BAAe,mBAAY,IA  
AZ,CAAf,EAakC,UAAIC,EAA8C,UAA9C,CADJ,GNp4BgG,sB8G3MM,oBxGklChF,IwGllCgF,C9G2MN,EMu4  
BpE,UNv4BoE,C;K;IM04BpG,mE;MAQoD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnG,OAAW  
,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAgB,UAAhB,EAA4B,CAA5B,EAA+B,UAA/B,EAakD,IAAID,CADJ,  
GNn5BgG,sBMs5B1E,MNt5B0E,EMs5BIE,UNT5BkE,C;K;IMy5BpG,mD;MAM+D,0B;QAAA,aAAsB,K;MACjF,  
OAAI,yBAAJ,GACI,sBAAQ,KAAR,UAA4B,UAA5B,KAA2C,CAD/C,GAGI,sBAAQ,KAAR,EAAe,CAAf,EAak  
B,gBAAIB,EAA0B,UAA1B,KAAyC,C;K;IAIjD,kD;MAMsD,0B;QAAA,aAAsB,K;MACxE,6BAAQ,IAAR,UAA2  
B,UAA3B,KAA0C,C;K;kFAE9C,4B;MAI0E,OAAA,KAAM,yBAAGB,SAAhB,C;K;IAM3C,yE;MACjC,oB;MAC  
A,8B;MACA,oB;MACA,kC;K;IAG8C,sF;MAAA,gE;MAC1C,iBAAqB,E;MACrB,yBAAwC,WAAX,yCAAW,EA  
AS,CAAT,EAAY,oCAAM,OAAIB,C;MACxC,uBAA2B,sB;MAC3B,gBAA0B,I;MAC1B,eAAmB,C;K;0EAEb,Y  
;MACI,IAAI,uBAAkB,CAAtB,C;QACI,iBAAY,C;QACZ,gBAAW,I;QAEX,IAAI,4CAAQ,CAAR,IAAa,uDAAa,y  
CAA1B,IAAmC,uBAAkB,yCAAM,OAA/D,C;UACI,gBAAW,qCAAyB,iBAAN,yCAAM,CAAzB,C;UACX,uBA  
AkB,E;UAEIB,YAAkB,iDAAN,yCAAM,EAAa,oBAAb,C;UACIB,IAAI,SAAS,IAAb,C;YACI,gBAAW,qCAAyB  
,iBAAN,yCAAM,CAAzB,C;YACX,uBAAkB,E;YAEIB,IAAK,QAAiB,KAAjB,aAAL,EAAY,SAAU,KAAY,a;Y  
ACZ,gBAAW,gCAAwB,KAAxB,C;YACX,yBAAoB,QAAQ,MAAR,I;YACpB,uBAAkB,0BAAwB,WAAU,CAAd  
,GAAiB,CAAjB,GAAwB,CAA5C,K;QAG1B,iBAAY,C;K;oEAIpB,Y;MAKiB,Q;MAJb,IAAI,mBAAa,EAajB,C  
;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,mE;MAEb,gBAAW,I;MACX,iBAAY,  
E;MACZ,OAAO,M;K;uEAGX,Y;MACI,IAAI,mBAAa,EAajB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;iDA9C5B  
,Y;MAA8C,+D;K;IAgEU,0E;MAAA,0C;QhB1mCjD,SgB2mCH,sBAAW,kBAAX,EAAuB,YAAvB,EAakD,kBA  
AID,C;QAAA,OAAwE,KAAK,CAAT,GAAY,IAAZ,GAAsB,OAAM,CAAN,C;O;K;IAIdG,iF;MAUkE,0B;QAAA,  
aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC7H,wBAAwB,KAAxB,C;MAEA,OAA  
O,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,gDAAjD,C;K;IAwBiD,gF;MAAA,0C;QAakB,Q;  
QAAA,oCAAU,sBAAV,EAA0B,YAA1B,EAAqD,kBAArD,EAAwE,KAAxE,aAAf,GAAG,UAAH,EAAe,WAA  
O,OAAtB,CAAtF,O;K;IAIB9E,mF;MAc0E,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAA  
A,QAAa,C;MACrI,wBAAwB,KAAxB,C;MACA,qBAAgC,OAAX,UAAW,C;MAEHc,OAAO,4BAAwB,SAAXB,E  
AA8B,UAA9B,EAA0C,KAA1C,EAAiD,sDAAjD,C;K;IAIX,wC;MnBlTCl,IAAI,EmBmtCl,SAAS,CnBntCb,CAAJ,  
C;QACI,cmBktCkB,8C;QnBjtClB,MAAM,gCAAyB,OAAQ,WAAjC,C;K;ImBkuCgE,sD;MAAA,qB;QAAE,yCA  
AU,EAAV,C;O;K;IAZhF,mE;MAWmE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACzG,OAAe,OAAt  
E,+BAakB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAL,iCAAJ,C;K;IAE1E,yD;MAWyD,0B;QAAA,  
aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC/F,IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,gBAAgB,WAAW,  
CAAX,C;QChB,IAAI,EAAC,SAh/BuC,YAAU,CAG/BID,CAAJ,C;UACI,OAAO,mBAAM,SAAN,EAAiB,UAAj  
B,EAA6B,KAA7B,C;;MAI2E,kBAAb,cAAtE,+BAakB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;Mb  
8OtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT  
,WAAy,WATgF,uBbsTIE,IatTKE,CbsThF,C;;MatThB,ObuTO,W;K;Ia5SmE,wD;MAAA,qB;QAAE,yCAAU,EAA  
V,C;O;K;IARhF,qE;MAOiE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACvG,OAAe,OAAtE,6BAAkB,  
UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAL,mCAAJ,C;K;IAE1E,2D;MAOuD,0B;QAAA,aAAsB,K;  
MAAO,qB;QAAA,QAAa,C;MAC7F,IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,OAAO,mBAAoB,oBAAd,  
WAAW,CAAX,CAAc,CAApB,EAAgC,UAAhC,EAA4C,KAA5C,C;;MAG+E,kBAAb,cAAtE,6BAAkB,UAAIB,U  
AA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;MbqNtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MA  
AA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,Wa7RgF,uBb6RIE,Ia7RkE,Cb6RhF,C;;Ma7RhB,Ob8RO,  
W;K;Ia3RX,0D;MASI,wBAAwB,KAAxB,C;MAEA,oBAAoB,C;MACpB,gBAAgB,sBAAQ,SAAR,EAAmB,aAA  
nB,EAakC,UAAIC,C;MACHB,IAAI,cAAa,EAAb,IAAmB,UAAAS,CAAhC,C;QACI,OAAO,OAAO,SAAK,WAAZ,  
C;;MAGX,gBAAgB,QAAQ,C;MACxB,aAAa,iBAAsB,SAAJ,GAAqB,eAAN,KAAM,EAAa,EAAb,CAArB,GAA2  
C,EAA7D,C;;QAET,MAAO,WA36B6E,8BA26B/D,aA36B+D,EA26BhD,SA36BgD,CAakC,WA26B/G,C;QACP,  
gBAAgB,YAAY,SAAU,OAAtB,I;QAEhB,IAAI,aAAa,MAAO,KAAP,MAAE,QAAQ,CAAR,IAAf,CAAjB,C;UAA  
2C,K;QAC3C,YAAY,sBAAQ,SAAR,EAAmB,aAAAnB,EAakC,UAAIC,C;;MACP,sBAAa,EAAb,C;MAET,MAAO

,WAI7BiF,8BAk7BnE,aAl7BmE,EAk7BpD,gBAI7BoD,CAAkC,Wak7BnH,C;MACP,OAAO,M;K;2EAGX,mC;MAOmD,qB;QAAA,QAAa,C;MAAmB,OAAA,KAAM,eAAM,SAAN,EAAY,KAAZ,C;K;+FAEzF,mC;MAU6D,qB;QAAA,QAAa,C;MAAuB,OAAA,KAAM,yBAAgB,SAAhB,EAAsB,KAAtB,C;K;IAEvG,iC;MAK2D,mCAAgB,MAAhB,EAawB,IAAxB,EAa8B,IAA9B,E;K;IAE3D,0B;MAKgD,OAAe,UAAf,uBAaE,C;K;IAqB/D,uD;MAQsB,Q;MAPIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAy,SAAL,SAAK,EAAO,KAAP,EAA2B,IAA3B,C;;MAGhB,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAL,KAAe,KAAM,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB;MAAIB,aAAU,CAAV,gB;QACI,IAAI,CAAS,SAAR,qBAAK,CAAL,CAAQ,EA AO,iBAAM,CAAN,CAAP,EAa8B,IAA9B,CAAb,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,6C;MAQsB,Q;M APIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAO,kBAAQ,KAAR,C;;MAGX,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I; MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAL,KAAe,KAAM,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB; MAAIB,aAAU,CAAV,gB;QACI,IAAI,qBAAK,CAAL,MAAW,iBAAM,CAAN,CAAF,C;UACI,OAAO,K;;;MAIf, OAAO,I;K;IAGX,oC;MAU+C,QAAM,SAAN,C;aAC3C,M;UAD2C,OACjC,I;aACV,O;UAF2C,OAEhC,K;;UACH ,MAAM,gCAAYB,mDAAGD,SAaZE,C;;K;IAGIB,0C;MAUsD,QAAM,SAAN,C;aACID,M;UADkD,OACxC,I;aA CV,O;UAFkD,OAEvC,K;;UAFuC,OAG1C,I;;K;I+Kr8Cz,sB;MAAA,0B;MAII,aAC+B,e;MAC/B,cACgC,e;MACH C,WAC6B,e;MAC7B,YAC8B,e;MAC9B,eACiC,e;MACjC,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,YAC8B,gB; MAC9B,aAC+B,gB;MAC/B,eACiC,gB;MACjC,iBACmC,gB;MACnC,qBAEuC,gB;MACvC,sBAEWc,gB;MACx C,kBACoC,gB;MACpC,cACgC,gB;MACHC,iBACmC,gB;MACnC,iBACmC,gB;MACnC,iBACmC,gB;MACnC,Y AC8B,gB;MAC9B,aAC+B,iB;MAC/B,aAC+B,iB;MAC/B,uBACyC,iB;MACzC,wBAC0C,iB;MAC1C,sBACwC,i B;MACxC,uBACyC,iB;MACzC,wBAC0C,iB;MAC1C,sBACwC,iB;MACxC,cACgC,iB;MACHC,oBACsC,iB;MA CtC,cACgC,iB;MACHC,gBACKC,iB;MACIC,aAC+B,iB;MAC/B,mBACqC,iB;MACrC,YAC8B,iB;MAC9B,UAC4 B,iB;MAC5B,mBACqC,iB;MACrC,gBACKC,iB;MACIC,mBACqC,iB;MACrC,sBACwC,iB;MAExC,sBAGwC,gB ;MAExC,uBAGyC,gB;K;;;IA7F7C,kC;MAAA,iC;QAAA,gB;;MAAA,0B;K;;;;2FCuE0C,Y;MAAQ,oCAAA,IA Ab,C;K;IAiBpB,yC;MAAQB,kB;K;mIAC3C,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;mIACnD,Y; MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAI B,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;mIACnD,Y;MACmD,OAAA,UAAM,Y AAN,aAAkB,CAAIb,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;mIACnD,Y;MACm D,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K; mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIb,C;K;qIACnD,Y;MACmD,OAAA,UAAM,YAAN,aA AkB,EAAlB,C;K;gDAEnD,Y;MAMoC,OAAA,UAAM,YAAY,iBAAQ,CAAR,EAaw,UAAM,YAAY,KAA7B,C; K;;;6EhEjH9D,yB;MAAA,iD;MAAA,4B;QAI4C,kBAAM,SAAN,C;O;KAJ5C,C;+EAMA,yB;MAAA,gD;MAAA, oC;QAI+D,kBAAM,SAAN,EAAY,MAAZ,C;O;KAJ/D,C;+EAMA,yB;MAAA,oC;MAAA,qC;QAIqE,sBAAM,SA AN,EAAY,OAAZ,C;O;KAJrE,C;IvIY4B,4B;MAmBxB,gC;MAnB6C,0B;MAW7B,UAEA,MAFA,EAGA,M;MAL Z,IkIjC8D,IIIiC9D,C;QACI,IAAI,kBAAJ,C;UACQ,mB;UAAJ,IAAI,sEAAsB,SAAtB,EAaj,C;YAAqC,MAAM,sB AAiB,YAAF,+CAAF,C;;UAEvC,qB;UAAJ,IAAI,0EAAuB,UAAvB,EAaj,C;YAAuC,MAAM,sBAaiB,YAAF,gD AAF,C;UACzC,qB;UAAJ,IAAI,kEAA+B,mBAA/B,CAAJ,C;YAAwD,MAAM,sBAaiB,YAAF,mCAAF,C;;;K;mF AZID,Y;MAAQ,kCAAA,CAAb,C;K;-FACU,Y;MAAQ,OAAA,eAAS,QAAT,GAAqB,C;K;qCACvE,Y;MAA0B,Q ADwB,eAAS,QAAT,GAAqB,CAC7C,MAAQB,C;K;sCAC/C,Y;MAA2B,QAFuB,eAAS,QAAT,GAAqB,CAE5C, MAAqB,C;K;yFACxB,Y;MAAQ,OAAl,kBAAJ,mF;K;IAahC,8B;MAAA,kC;MACI,YAC4B,gB;MAE5B,gBACgC ,iBAaiB,UAAjB,C;MACHC,4BAAsC,uC;K;mDAEtC,yC;MAGI,2BAAoB,KAApB,EAA2B,UAA3B,EAauC,UA AvC,C;K;iJAM8B,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;iJAIC,yB;MAAA,6C;MAAA,iD; MAAA,4B;QAAQ,sD;O;KAAR,C;iJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;mJAKF,y B;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ, uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAKH,yB;MAAA,6C;MA AA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;m JAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;yIAKR,yB;MAAA,6C;MAAA,iD;MAAA,4B; QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C ;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR, C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4

B;QAAQ,kD;O;KAAR,C;qIAKL,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;qIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;qIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;mIAKJ,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;uDAK9B,iB;MAK+C,OAAM,WAAN,KAAM,yC;K;uDAErD,iB;MAKgD,OAAM,aAAN,KAAM,yC;K;uDAEtD,iB;MASKD,OAAM,aAAN,KAAM,yC;K;wDAGxD,iB;MAKgD,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAAN,KAAM,0C;K;wDAGzD,iB;MAKgD,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAAN,KAAM,0C;K;mDAGzD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAEID,iB;MAS8C,OAAM,aAAN,KAAM,qC;K;mDAGpD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAEID,iB;MAS8C,OAAM,aAAN,KAAM,qC;K;iDAGpD,iB;MAKyC,OAAM,WAAN,KAAM,mC;K;iDAE/C,iB;MAK0C,OAAM,aAAN,KAAM,mC;K;iDAEhD,iB;MAS4C,OAAM,aAAN,KAAM,mC;K;gDAGID,iB;MAKwC,OAAM,WAAN,KAAM,kC;K;gDAE9C,iB;MAKyC,OAAM,aAAN,KAAM,kC;K;gDAE/C,iB;MAS2C,OAAM,aAAN,KAAM,kC;K;iDAEjD,iB;;QAY4C,OACxC,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,sCAAmC,KAAAnC,OAAzB,EAASe,CAAtE,C;;UAHkC,O;;K;0DAM5C,iB;;QAMqD,OACjD,cAAc,KAAAd,EAAiC,IAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,0CAAuC,KAAvC,OAAzB,EA A0E,CAA1E,C;;UAH2C,O;;K;uDAMrD,iB;;QAWmD,OAC/C,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UAFiD,OAG/C,I;;UAH+C,O;;K;gEAMnD,iB;;QAK4D,OACxD,cAAc,KAAAd,EAAiC,IAAjC,C;;QACF,+C;UAF0D,OAGx D,I;;UAHwD,O;;K;;IA/XhE,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;oCAwYA,Y;MAC6C,kBAAy,YAAD,aAA X,EAzZK,eAAS,QAAT,GAAqB,CAyZ1B,C;K;qCAE7C,iB;MAiBW,Q;MATH,IAAA,IAAK,aAAL,C;QACI,IAAI ,KAAM,WAAN,IAAqB,IAAK,WAAL,KAakB,KAAM,WAAxB,gBAAoC,CAA7D,C;UACI,OAAO,I;;UAEP,MA AM,gCAAyB,2EAazB,C;WAEd,IAAA,KAAM,aAAN,C;QAASB,OAAO,K;MAI7B,KA7a0C,eAAS,QAAT,GAAq B,CA6a/D,OAA0B,KA7agB,WAAS,QAAT,GAAqB,CA6a/D,E;QACI,aAAa,IAAK,QAAL,KAAa,KAAM,QAAAnB ,C;QAET,uB;UACI,iCAA0B,MAA1B,C;;UAEA,kCAA2B,MAA3B,C;aAGZ,IAAA,IAAK,eAAL,C;QACI,mCAA qB,IAAK,QAA1B,EAAiC,KAAM,QAAvC,C;;QAEA,mCAAqB,KAAM,QAA3B,EAAkC,IAAK,QAAvC,C;MAB R,W;K;gDAiBJ,kC;MAGW,Q;MAFP,kBAakB,cAAc,UAAAd,C;MACIB,mBAAmB,eAAa,WAAb,C;MACZ,IAAI, 8EAAS,C,mBAAtC,CAAJ,C;QACH,yBAAYB,oBAAa,cAAc,WAAAd,CAAb,C;QACzB,uBAAGB,cAAc,YAAAd,MA A8B,kBAA9B,CAAhB,C;;QAEA,wBAA8B,WAAb,YAAa,yBAASB,UAAtB,CAA9B,C;;MAJJ,W;K;sCAQJ,iB;M AMuD,wBAAS,KAAD,aAAR,C;K;uCAEvD,iB;MAQe,UAUJ,M;MAXP,IAAI,iBAAJ,C;QAEQ,cAAS,CAAT,C;U AAAC,MAAM,gCAAyB,mEAazB,C;aACpB,YAAQ,CAAR,C;UAAa,W;;UACL,OAAO,IAAD,a;QAHZ,W;;MAMJ, IAAI,UAAS,CAAb,C;QAAgB,OAAO,qC;MAEvB,YAAY,Y;MACZ,aAAa,mCAAQ,KAAR,E;MACN,IAAI,kBA AJ,C;QACH,IAAI,yEAAJ,C;UAEI,yBAAGB,MAAhB,C;;UAEA,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;Y ACI,mCAA0B,MAA1B,C;;YAEA,aAAa,cAAc,KAAAd,C;YACb,eAAe,eAAQ,cAAc,MAAd,CAAR,C;YACf,mBA AmB,oCAAS,KAAT,E;YACnB,kBAakB,iBAAe,cAAc,sCAAW,KAAX,EAAd,CAAF,C;YACIB,IAAI,4CAAE,KA Af,IAAwB,MAAxB,KAakC,gBAAgB,YAAhB,gBAAgC,CAAtE,C;cACI,0BAA6B,WAAZ,WAAy,EAAS,8BAA a,UAAb,CAAT,CAA7B,C;;cAEA,SAAL,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAANB,IAA0B ,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;;;QAK3D,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;UACI,0BAAwB, WAAP,MAAO,EAAS,8BAAa,UAAb,CAAT,CAAxB,C;;UAEA,SAAL,YAAM,WAAN,KAAM,CAAN,EAAMB,W AAN,KAAM,CAANB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;MAvBvD,a;K;uCA4BJ,iB;MASI,eAAqB,W AAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAA3B,C;QACI,OAAO,mBAAM,QAAN,C;;MAGX,WAAW,kB; MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;qCAGIB,iB;MAQe,Q;M ADX,IAAI,UAAS,CAAb,C;QAEQ,sB;UAAgB,gD;aAChB,sB;UAAgB,4D;;UACR,MAAM,gCAAyB,4DAAzB,C; QAHIB,W;;MAMJ,IAAI,kBAAJ,C;QACI,OAAO,gBAAgB,qCAAQ,KAAR,EAAhB,C;;QAEP,IAAI,iBAAJ,C;UA CI,OAAO,mBAAa,WAAN,KAAM,CAAb,C;QAEX,aAAa,qCAAQ,KAAR,E;QAEb,IAAI,kEAAGC,mBAAhC,CA AJ,C;UACI,UAAU,cAAc,sBAAS,oCAAS,KAAT,EAAT,CAAd,0BAA0C,KAA1C,E;UACV,OAAO,gBAAgB,cA Ac,MAAd,MAAwB,GAAxB,CAAhB,C;;QAEX,OAAO,iBAAiB,MAAjB,C;;K;qCAIf,iB;MAOI,eAAqB,WAAN,K AAM,C;MACrB,IAAa,QAAT,KAAuB,KAAvB,IAAgC,aAAY,CAAhD,C;QACI,OAAO,iBAAI,QA AJ,C;;MAGX, WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;oCAGIB,iB;

MAEI,kBAaKb,SAAM,IAAK,cAAX,EAaWb,KAAM,cAA9B,C;MACIB,OAAO,IAAK,kBAAS,WAAT,CAAL,G  
AA6B,KAAM,kBAAS,WAAT,C;K;oCAG9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,oCAAW,C;K;oCAE  
9C,Y;MACmC,+BAAy,yCAAS,WAArB,KAAiC,wBAAy,qDAAa,WAAzB,C;K;kCAEpE,Y;MACiC,QAAC,iB;K  
;yFAGC,Y;MAAQ,OAAI,iBAAJ,GAAMb,IAAD,aAAIB,GAA6B,I;K;yCAExE,iB;MACI,kBAaKb,IAAK,WAAL,  
KAAkB,KAAM,WAAxB,C;MACIB,IAAI,yBAAC,CAAd,IAAmB,CAAA,WAAy,QAAZ,GAAwB,CAAxB,MAA6  
B,CAApD,C;QACI,OAAO,IAAK,WAAS,iBAAU,KAAM,WAAhB,C;MAEzB,QAAQ,CA11BsC,eAAS,QAAT,GA  
AqB,CA01B3D,KAAyB,KA11Ba,WAAS,QAAT,GAAqB,CA01B3D,K;MACR,OAAW,iBAAJ,GAaKb,CAAC,CA  
AD,IAAIB,GAA0B,C;K;uHAMrC,kB;MAeI,OAAO,OAAO,gBAAP,EAAoB,mBAAPb,EAAoC,qBAAPc,EAAsD,  
qBAAtD,EAAwE,yBAAXE,C;K;uHAGX,kB;MAcI,OAAO,OAAO,iBAAP,EAAqB,qBAArB,EAAuC,qBAAvC,EA  
AyD,yBAAzD,C;K;uHAGX,kB;MAaI,OAAO,OAAO,mBAAP,EAAuB,qBAAvB,EAAYC,yBAAzC,C;K;uHAGX,  
kB;MAYI,OAAO,OAAO,mBAAP,EAAuB,yBAAvB,C;K;0FAKP,Y;MAAQ,OAAI,iBAAJ,GAaKb,CAAIB,GAA0  
B,6CAAE,EAaf,EAAMb,Q;K;4FAIrD,Y;MAAQ,OAAI,iBAAJ,GAaKb,CAAIB,GAA0B,+CAAiB,EAajB,EAaq  
B,Q;K;4FAIvD,Y;MAAQ,OAAI,iBAAJ,GAaKb,CAAIB,GAA0B,+CAAiB,EAajB,EAaqB,Q;K;gGAIvD,Y;MAC  
I,sB;QADI,OACY,C;WACHb,wB;QAFI,OAey,cAAc,wCAAQ,IAAR,EAAd,CAA6B,Q;QAFzC,OAGK,wCAAQ,  
UAAR,EAAuB,Q;K;0CAMxC,gB;MAQiB,UAAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;QAA4B,SAA  
P,wCAAO,kB;WAC5B,iBAAA,qDAAa,WAAb,E;QAAgC,SAAP,wCAAO,kB;QAG5B,6BAAoB,YAAM,WAA1  
B,EAAsC,kBAAtC,EAAMd,IAAnD,C;MALR,a;K;wCAUJ,gB;MAUiB,UAAN,M;MAAM,sB;MACT,iBAAA,yC  
AAS,WAAT,E;WACA,iBAAA,qDAAa,WAAb,E;QACQ,+BAAoB,YAAPb,EAA2B,kBAA3B,EAAwC,IAAxC,  
C;MAHZ,a;K;uCAOJ,gB;MAUI,OAAa,WAAb,oBAAO,IAAP,CAAA,4BAAyD,Q;K;kFAKhD,Y;MAAQ,6D;K;mf  
AKP,Y;MAAQ,8D;K;qFAKN,Y;MAAQ,gE;K;qFAKR,Y;MAAQ,gE;K;0FAKH,Y;MAAQ,qE;K;0FAKR,Y;MAA  
Q,qE;K;yFAKT,Y;MAAQ,oE;K;uFASrC,Y;MAAQ,2D;K;wFAQR,Y;MAAQ,4D;K;0FAQR,Y;MAAQ,8D;K;0FA  
QR,Y;MAAQ,8D;K;+FAQR,Y;MACI,OAAW,uBAAGb,eAApB,GAAgC,YAAhC,GAA2C,4D;K;+FAAtD,Y;MAA  
Q,mE;K;8FAYR,Y;MAEW,Q;MADP,YAAY,Y;MAER,uB;QAaE,Y;WACf,8C;WACA,+C;QACQ,qBAAC,KAA  
d,C;MAJZ,W;K;2CAUR,Y;MASuC,8B;K;4CAEvC,Y;MASwC,+B;K;kCAExC,Y;MAuBwC,Q;MAAA,sB;MACp  
C,qB;QAD8B,OACxB,I;WACN,iBAAA,yCAAS,WAAT,E;QAF8B,OAET,U;WACrB,iBAAA,qDAAa,WAAb,E;Q  
AH8B,OAGL,W;QAErB,iBAAiB,iB;Q8HzhBF,gBAAhB,sB;Q9H2hBK,e;UAGb,yBAAO,EAAP,C;QACF,YAA  
d,kB;QA9RD,WAAO,iB;QAAP,YAAoB,oB;QAAPb,cAAoC,sB;QAAPc,cAAAsD,sB;QAAtD,kBAAwE,0B;QAsS/  
D,0B;QAPJ,cAAc,iB;QACd,eAAe,UAAS,C;QACxB,iBAAiB,YAAW,C;QAC5B,iBAAiB,YAAW,CAAX,IAAgB,  
gBAaE,C;QACHd,iBAAiB,C;QACjB,IAAI,OAaj,C;UACI,yBAAO,IAAP,CAAA,gBAAO,GAAP,C;UACb,+B;Q  
AEJ,IAAI,aAAa,YAAY,cAAc,UAA1B,CAAb,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;  
UACtB,yBAAO,KAAP,CAAc,gBAAO,GAAP,C;QAEIB,IAAI,eAAe,eAAe,YAAY,OAA3B,CAAF,CAAJ,C;UACI  
,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,OAAP,CAAgB,gBAAO,GAAP,C;QAEpB,I  
AAI,UAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UAEIB,gBAAW,CAAX,IAAgB,OAAhB,I  
AA2B,QAA3B,IAAuC,UAAvC,C;YACI,mCAAiB,OAajB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C,GAA1C,E  
AA2D,KAA3D,C;eACJ,mBAaE,OAaf,C;YACI,mCAAiB,cAAc,OAAd,IAAjB,EAA0C,cAAc,OAAXD,EAAMe,C  
AAAnE,EAAsE,IAATe,EAaWf,KAAxF,C;eACJ,mBAaE,IAaf,C;YACI,mCAAiB,cAAc,IAAd,IAAjB,EAAsC,cAA  
c,IAApD,EAA2D,CAA3D,EAA8D,IAA9D,EAAGf,KAAhF,C;YAEA,yBAAO,WAAP,CAAoB,gBAAO,IAAP,C;;  
QAGhC,IAAI,cAAc,aAAa,CAA/B,C;UAAkC,yBAAO,CAAP,EAAU,EAAV,CAAE,gBAAO,EAAP,C;QAvC/B,O  
Ox1B3B,SuHoUqC,W;;K;4C9HikB5C,yE;MACI,yBAAO,KAAP,C;MACA,IAAI,eAAc,CAAIB,C;QACI,yBAAO,  
EAAP,C;QACA,iBAAuC,WAAtB,UAAW,WAAW,EAAS,cAAT,EAAYB,EAAzB,C;QACR,sB;;UsB5zBzB,Q;UA  
AA,OAAQ,WAAR,etB4zBc,UsB5zBd,CAAQ,CAAR,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,ItB2zBiD,UsB3  
zBnC,YtB2zBU,UsB3zBV,YAAK,KAAAL,EtB2zBmC,MAAM,EsB3zBvD,C;cACI,qBAAO,K;cAAP,uB;;UAGR,q  
BAAO,E;;QtBuzBC,oBAAoB,qBAAuC,CAAvC,I;QAEhB,KAAC,SAAD,IAAc,gBAAGb,CAA9B,C;UAAmC,8B  
AAY,UAAZ,EAaWb,CAAxB,EAA2B,aAA3B,C;;UAC3B,8BAAy,UAAZ,EAaWb,CAAxB,EAA2B,CAAC,CAA  
C,gBAAGb,CAAhB,IAAD,IAAsB,CAAtB,IAAD,IAA4B,CAA5B,IAA3B,C;;MAGhB,yBAAO,IAAP,C;K;0CAGJ,  
0B;MAGbWc,wB;QAAA,WAAgB,C;MIn9BxD,IAAI,EJo9BQ,YAAY,CIp9BpB,CAAJ,C;QACI,cJm9ByB,oD;QII  
9BzB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MJm9BN,aAAa,sBAAS,IAAT,C;MACb,IAAW,WAAP,MAAO,CAA  
X,C;QAAYB,OAAO,MAAO,W;MACvC,OAAO,sBAAsB,MAAtB,EAAuC,eAAT,QAAS,EAAa,EAAb,CAAvC,IA



AgE,UAAL,IAAK,C;K;qCAI3E,Y;M8HvmBuB,gBAAhB,sB;M9HqnBH,IAAI,iBAAJ,C;QAAkB,yBAAO,EAAP,C;MACIB,yBAAO,IAAP,C;MAC4B,YAAAd,kB;MAxWP,YAAO,kB;MAAP,cAAqB,sB;MAArB,cAAuC,sB;MAAvC,kBAAYD,0B;MAyW5D,cACY,K;MACZ,IAAI,iBAAJ,C;QAEI,wB;;MAEJ,eAAe,oB;MACf,iBAAiB,YAAW,C AAX,IAAgB,gBAAe,C;MACHd,iBAAiB,YAAW,CAAX,KAAiB,cAAc,QAA/B,C;MACjB,IAAI,QAAJ,C;QACI,y BAAO,OAAP,CAAc,gBAAO,EAAP,C;;MAEIB,IAAI,UAAJ,C;QACI,yBAAO,OAAP,CAAgB,gBAAO,EAAP,C;; MAEpB,IAAI,eAAe,CAAC,QAAD,IAAa,CAAC,UAA7B,CAAJ,C;QACI,mCAAiB,OAAjB,EAA0B,WAA1B,EA AuC,CAAvC,EAA0C,GAA1C,EAA2D,IAA3D,C;;MApBuB,OOx7B5B,SuHoUqC,W;K;,,,;kC9H5YhD,Y;MAAA, c;MAuBiD,2D;MAvBjD,a;K;gCAAA,iB;MAAA,2IAuBiD,gDAvBjD,G;K;IA8hCA,qC;MAIW,Q;MAAA,IAAI,6D AAJ,C;QACH,uBAAgB,4BAAiC,oBAAL,SAAK,CAAjC,EAA2C,IAA3C,yCAAhB,C;;QAES,oBAAT,8BAAS,EA AW,IAAX,C;MAHb,W;K;IAMJ,uC;MAII,kBAAkB,4BAA4B,SAA5B,0CAAiE,IAAjE,C;MACIB,IAAa,WAAD,a AAR,yDAAsB,WAAtB,CAAJ,C;QACI,OOAO,gBAAgB,4BAA4B,SAA5B,EAakC,IAAIc,yCAAhB,C;;QAEp,aA Aa,sBAAoB,SAApB,EAA0B,IAA1B,0C;QACb,OOAO,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;I AIf,uC;MAaW,Q;MAHP,gBAAgB,oBAAoB,SAApB,EAA0B,IAA1B,yC;MiviChB,IAAI,CJwiCI,CAAW,QAAV, SAAU,CiXiCnB,C;QACI,cJuiC0B,+B;QItiC1B,MAAM,gCAAYB,OAAQ,WAAjC,C;;MJuiCV,YAAAsB,YAAV,SA AU,C;MACf,IAAI,sEAAqB,SAArB,CAAJ,C;QACH,uBAAgB,KAAhB,C;;QAEA,aAAwE,YAA3D,oBAAoB,SAA pB,EAA0B,IAA1B,0CAA2D,C;QACxE,kCAA2B,MAA3B,C;;MAJJ,W;K;IAGBuB,oC;MAAQ,oE;K;IAOP,sC;MA AQ,sE;K;IAWN,sC;MAAQ,sE;K;IAQV,qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQX, qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQhB,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K ;IAWN,kC;MAAQ,kE;K;IAQX,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQb,8B;MAA Q,8D;K;IAOP,gC;MAAQ,gE;K;IAWN,gC;MAAQ,gE;K;IAQZ,6B;MAAQ,6D;K;IAOP,+B;MAAQ,+D;K;IAWN,+ B;MAAQ,+D;K;yEAG/B,+B;MAIqE,8BAAW,SAAX,C;K;2EAERe,+B;MAUwE,8BAAW,SAAX,C;K;IAIxE,yC; MACI,aAAa,KAAM,O;MACnB,IAAI,WAAU,CAAd,C;QAAiB,MAAM,gCAAYB,qBAAZB,C;MACvB,YAAY,C; MACZ,aAAa,gCAAS,K;MACTb,qBAAqB,U;MACrB,QAAM,iBAAM,KAAN,CAAN,C;aACI,E;aAAA,E;UAAy, qB;UAAZ,K;;MAEJ,cAAc,QAAQ,C;MACTb,iBAAiB,WAAiB,aAAN,KAAM,EAAW,EAAX,C;MAE9B,cAAU,K AAV,C;QACI,MAAM,gCAAYB,eAAzB,C;WACV,qBAAM,KAAN,MAAgB,EAhB,C;QACI,IAAI,mCAAW,M AAF,C;UAAuB,MAAM,+B;QAC7B,sBAAsB,K;QACTb,sBAAsB,K;QACTb,eAA8B,I;QAC9B,OOAO,QAAQ,MA Af,C;UACI,IAAI,iBAAM,KAAN,MAAgB,EAAPB,C;YACI,IAAI,mBAAMB,mCAAW,MAAIC,C;cAA0C,MAA M,+B;YACHd,kBAAkB,I;YACIB,Q;;UAekB,iBAAe,K;UA+EjD,QAHGc,U;UAIhC,Y;YAAO,eAhFqB,KAgFjB, O;YAAJ,S;cAAc,SAAU,YAhFH,KAgFG,YAAK,CAAL,E;cAAV,OAhFqC,CAAM,kBAAK,EAAL,CAAN,qCAA kB,2C;,,,;YAgFnC,a;;UAhF7B,gBAAgB,KiBv1CgE,WjBmqClF,UiBnqCkF,EjBwqCrF,CiBxqCqF,C;UjBwlChF,IA AI,SuBrhCgC,YAAU,CvBqhC9C,C;YAAyB,MAAM,+B;UAC/B,gBAAS,SAAU,OOAnB,I;UACqB,cAAU,K;UsB zrCpC,U;UAAA,IAAI,WAAS,CAAT,IAAc,WAAS,iBtByrCP,KsBzrCO,CAA3B,C;YAAA,StByrCoB,KsBzrCkB, YAAI,OOAJ,C;;YtByrCO,MAAM,gCAAYB,qCAAzB,C;;UAA9C,qB;UACA,qB;UACA,WAAW,sBAAsB,QAAtB ,EAAgC,eAAhC,C;UACX,IAAI,YAAy,IAAZ,IAAoB,yBAAy,IAAZ,MAAxB,C;YAA0C,MAAM,gCAAYB,yCA AzB,C;UACHd,WAAW,I;UACX,eAAyB,WAAV,SAAU,EAAQ,EAAR,C;UACzB,IAAI,+CAAgC,WAAW,CAA/ C,C;YACI,YAAy,SiBjmCgE,WjBimC5C,CiBjmC4C,EjBimCzC,QiBjmCyC,C;YjBkmC5E,4BAA2C,aAAjC,0BA A0B,KAA1B,CAAiC,EAAW,IAAX,CAA3C,C;YACA,4BAAMd,aAAX,SAA9B,SiBtmCmD,WjBsmC/B,QiBtmC +B,CjBsmCrB,CAAW,EAAW,IAAX,CAAnD,C;;YAEA,4BAA+C,aAArC,0BAA0B,SAA1B,CAAqC,EAAW,IAA X,CAA/C,C;;aAIZ,c;QACI,MAAM,+B;;QACV,IAAM,cAAN,KAAM,EAAC,KAAd,EAAqB,cAArB,EAAqC,CA ArC,EQ/xCH,MAAO,KR+xCmD,SAAS,KAAT,IQ/xCnD,ER+xCmE,cAAe,OQ/xCIF,CR+xCJ,EAA4G,IAA5G,CA AN,C;UACI,SAAS,gCAAS,S;;UAiIB,iBAA8B,I;UAC9B,iBAAiB,K;UACjB,kBAAkB,CAAC,O;UACnB,IAAI,W AAW,iBAAM,KAAN,MAAgB,EAA3B,IAAwC,QAAN,KAAM,CAAN,KAAgB,EAAtD,C;YACI,cAAc,I;YACd,I AAI,oCAAW,uBAAX,EAAW,MAAX,CAAJ,C;cAAyB,MAAM,gCAAYB,eAAzB,C;;UAEnC,OOAO,QAAQ,MA Af,C;YACI,IAAI,cAAc,WAAIB,C;cA8CZ,UA7CwC,K;cA8CxC,Y;gBAAO,mBA9CiB,KA8Cb,O;gBAAJ,W;kBA Ac,SA9C4B,UA8CIB,YA9CP,KA8CO,YAAK,GAAL,EA9CkB,MAAM,E;,,,;gBA8Cd,iB;;cA9CzB,QA+CT,G;;YA 7CK,aAAa,I;YACS,mBAAe,K;YA0CjD,UAHGc,Y;YAIhC,Y;cAAO,mBA3CqB,KA2CjB,O;cAAJ,W;gBAAc,WA AU,YA3CH,KA2CG,YAAK,GAAL,E;gBAAV,SA3CqC,CAAM,kBAAK,EAAL,CAAN,uCAAkB,oBAAM,E;,,,;c A2CzC,iB;;YA3C7B,kBAAgB,KiB5nCGE,WjBmqClF,YiBnqCkF,EjBwqCrF,GiBxqCqF,C;YjB6nChF,IAAI,WuB

IjCgC,YAAU,CvB0jC9C,C;cAAyB,MAAM,+B;YAC/B,gBAAS,WAAU,OAA nB,I;YACqB,mBA Ae,K;Y AuChD,UAHgC,Y;YAIhC,Y;cAAO,mBAxCoB,KAwChB,O;cAAJ,W;gBAAc,WAAU,YAxCJ,KAwCI,YAAK,GAAL,E;gBAAV,SAxCoC,CAAM,kBA AK,GAAL,CAAN,mC;;;cAwChB,iB;;YAxC7B,eAAe,KiB/nCiE,WjBmqClF,YiBnqCkF,EjBwqCrF,GiBxqCqF,C;YjBgoChF,gBAAS,QAAS,OAAIB,I;YACA,aAAW,wBAAwB,QAAXB,C;YACX,IAAI,cAA Y,IAAZ,IAAoB,2BAAY,MAAZ,MAAxB,C;cAAOC,MAAM,gCAAyB,yCAAzB,C;YAC hD,aAAW,M;YACX,iBAAYB,WAAV,WAAU,EAAQ,EAAR,C;YACzB,IAAI,aAAW,CAAf,C;cACI,cAA Y,WiBtoCgE,WjBsoC5C,CiBtoC4C,EjBsoCzC,UiBtoCyC,C;cjBuoC5E,4BAAYB,aAAT,OAAN,OAAM,CAAS,EAAW,MAAX,CAAzB,C;cACA,4BAAMd,aAAX,SAA9B,WiB3oCmD,WjB2oC/B,UiB3oC+B,CjB2oCrB,CAAW,EAAW,MAAX,CAAnD,C;cACA,IAAI,QAAQ,MAAZ,C;gBAAoB,MAAM,gCAAyB,mCAAzB,C;;cAE1B,4BAA6B,aAAT,OA AV,WAAU,C AAS,EAAW,MAAX,CAA7B,C;;;MAKhB,OAAW,UAAJ,GA AiB,MAAD,aAAhB,GAA6B,M;K;IAIxC,0C;MACI,aAAa,KAAM,O;MACnB,iBA AiB,C;MACjB,IAAI,SAAS,CAAT,IAAc,YAAY,IAAZ,mBAAM,CAAN,EAAIB,C;QAAoC,+B;;MACHc,YAAC,SAAS,UAA T,IAAD,IAAwB,E;MAAxB,S;QAA4D,gBAA7B,yBA AkB,iBAAN,KAAM,CAAIB,C;QAA6B,c;;UU4ThD,U;UADhB,IAAI,wCAAsB,mBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CV5T4C,CAAa,kBA AK,EAAL,CAAb,oCU4TjC,OV5TiC,EU4ThD,C;cAAyB,aAAO,K;cAAP,e;;UAC/C,aAAO,I;;QV7TyD,iB;;MAAhE,S;QAEI,OAAW,iBAAM,CAAN,MAAY,EAAhB,sD;;MAGX,OAAiB,WAAN,KAAM,EAAW,GAAX,CAAV,GAAYC,OAAR,QAAN,KAAM,EAAK,CAAL,CAAQ,CAAzC,GAA6D,OAAN,KAAM,C;K;IAKxE,0D;MAII,QAHgC,U;MAIhC,OAAO,IAAI,gBAAJ,IAJqC,SAIvB,CAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAYC,a;;MAJzC,OiBnqC4F,oBjBmqClF,UiBnqCkF,EjBwqCrF,CiBxqCqF,C;K;IjBqqChG,qD;MACI,QAAQ,U;MACR,OAAO,IAAI,gBAAJ,IAAc,UAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAYC,a;;MACzC,OAAO,C;K;;;IAMBX,8B;MAA+C,qCAAQ,OAAR,E;K;IAC/C,+B;MAAgD,2CAAS,OAAT,E;K;IAEhD,sC;MAAiD,oBAAS,sBAAGB,CAAhB,CAAT,C;K;IACjD,wC;MAAMd,oBAAU,uBAAiB,CAAjB,CAAD,yBAAuB,CAAvB,EAAT,C;K;IACnD,oD;MAAoE,oBAAU,sBAAGB,CAAhB,CAAD,yBAA sB,iBAAtB,EAAT,C;K;IACpE,0C;MACI,IAAI,sEAAqB,SAArB,CAAJ,C;QAAA,OACI,gBAAGB,KAAhB,C;;QADJ,OAGI,iBA AiB,cAAc,KAAd,CAAjB,C;;K;IAGR,4C;MACI,IAAI,kEAAgC,mBAAhC,CAAJ,C;QAAA,OACI,gBAAGB,cAAc,MAAd,CAAhB,C;;QADJ,OAGI,iBAAwB,WAAP,MAAO,yBAAsB,UAA tB,CAAxB,C;;K;IwMI3CR,8B;MAEgD,QAAM,SAAN,M;aAC5C,a;UAD4C,OACHB,I;aAC5B,c;UAF4C,OAEf,I;aAC7B,c;UAH4C,OAGf,I;aAC7B,S;UAJ4C,OAIpB,G;aACxB,S;UAL4C,OAKpB,G;aACxB,O;UAN4C,OAMtB,G;aACtB,M;UAP4C,OA OvB,G;;UpMuEwB,MAAM,6BAA8B,CoMtEnE,mBAAGB,SpMsEmD,YAA9B,C;;K;IoMnEvD,4C;MACwE,QAAM,SAAN,C;aACpE,I;UADoE,6C;aAEpE,I;UAFoE,8C;aAGpE,I;UAHoE,8C;aAIpE,G;UAJoE,yC;aAKpE,G;UALoE,yC;aAMPe,G;UANoE,uC;aAOpE,G;UAPoE,sC;;UAQ5D,MAAM,gCAAyB,uCAAoC,SAA7D,C;;K;IAGlB,yD;MAGQ,KAAC,eAAD,C;QAEQ,IADE,OACF,Q;UAHZ,sC;;UAIoB,MAAM,gCAAyB,4EAAqD,OAARd,CAAzB,C;;QAIIB,QAAM,OAAN,C;eACI,E;YATZ,uC;eAU Y,E;YAVZ,yC;eAWY,E;YAXZ,yC;;YAYoB,MAAM,gCAAyB,yDAAkC,OAAIC,CAAzB,C;;;K;IC5F9B,4B;K;;;MC4BI,kC;;IAXA,gC;MAAA,oC;MAM0B,2BAAc,iC;K;8CACpC,Y;MAAkC,OAAA,iCAAoB,W;K;6CADhC,Y;MAAA,yC;K;;IAN1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAWA,gC;MAAA,oC;K;;IAAA,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;;IAKJ,oB;K;qCAcI,oB;MAK8D,4BAAiB,IAAjB,EAAuB,QAAvB,C;K;sCAE9D,oB;MAK+D,wBAAM,QAAD,aAAL,C;K;sCAG/D,Y;MAMqC,QAAC,iBAAa,a;K;yCAEnD,Y;MAMwC,OAAA,iBAAa,a;K;;4EAIzD,yB;MAAA,4C;MAAA,mC;QAQuE,MAAM,WAAM,0BAAN,C;O;KAR7E,C;mFAUA,yB;MAAA,4C;MAAA,mC;QAQsE,MAAM,WAAM,0BAAN,C;O;KAR5E,C;IAY8B,4C;MAAiD,mB;MAAhD,gB;MAAoB,4B;K;4CAC/C,Y;MAAsC,OAAA,SAAK,aAAL,cAAoB,eAApB,C;K;6CAEtC,oB;MAAkD,4BAAiB,SAAjB,EAAuB,4BAAa,QAAb,CAAvB,C;K;;IChGV,sC;MAAC,gB;K;IAOf,4E;MAA8G,mB;MAA7G,4B;MAA6B,8B;MAAgD,sB;K;+DACpG,Y;MAAsC,OAAgC,aAA/B,iBAAW,OAAX,UAAoB,gBAApB,CAA+B,EAAW,iBAAW,KAA tB,CAAhC,cAA8D,aAA9D,C;K;gEACtC,oB;MAAkD,+CAAa,gBAAb,EAAwB,iBAAXB,EAAoC,0BAAS,QAAT,CAApC,C;K;;+CAGtD,Y;MAAmC,+CAAa,WAAb,EAAqB,IAArB,EAA2B,gCAAS,KAApC,C;K;;IAUO,wC;MAAC,gB;K;IAOf,gF;MAAkH,mB;MAAjH,4B;MAA+B,8B;MAAkD,sB;K;mEAC1G,Y;MAAsC,OAAgC,aAA/B,iBAAW,OAAX,GAAoB,gBAAW,EAAW,iBAAW,KAA tB,CAAhC,cAA8D,aAA9D,C;K;oEACtC,oB;MAAkD,mDA Ae,gBAAf,EAA0B,iBAA1B,EAA sC,0BAAS,QAAT,CAAtC,C;K;;iDAGtD,Y;MAMmC,mDA Ae,WAAf,EAAuB,IAAvB,EAA6B,gCAAS,KAA tC,C;K;;IAGvC,0B;MAGB8B,yE;MAC1B,mB;K;oCAEA,Y;MAA4B,qB;K;iDAE5B,oB;MAWc,Q;MADV,gBAAGB,QAAS,gBAAO,SAAP,C;MACf,IAAI,gDAA+B,4

CAAnC,C;QAEN,iBAaIB,mBAAU,SAAV,C;QACjB,IAAI,mBAAY,SAAZ,gBAAYB,CAAZB,IAA8B,mBAAY,U  
AAZ,eAAyB,CAA3D,C;UAA8D,gBAAS,QAAT,C;QAC9D,iB;;QAEA,YAAY,QAAS,kBAAS,SAAT,C;QAERb,m  
BAAiB,4BAAU,K;QAC3B,IAAI,sDAA+B,kDAAAnC,C;UAAgE,gBAAS,QAAT,C;QACrD,8BAAX,YAAW,C;;M  
AVf,qB;K;0CacJ,oB;MACI,MAAM,6BAAsB,iDAA+C,cAA/C,qCAA0E,QAA1E,MAAtB,C;K;;qFC7Fd,yB;MAA  
A,yC;MAAA,wB;QA2BI,WAAW,8B;QAhB6B,KAiBxC,E;QAJBA,OAKBO,IAAK,a;O;KA7BhB,C;uFAeA,4B;M  
AYI,WAAW,mB;MACX,O;MACA,OAAO,IAAK,a;K;IAYe,qC;MAAC,kB;MAAc,wB;K;;sCAR9C,Y;MAQgC,iB  
;K;;sCARhC,Y;MAQ8C,oB;K;wCAR9C,2B;MAAA,sBAQgC,qCARhC,EAQ8C,8CAR9C,C;K;oCAAA,Y;MAAA,  
OAOgC,iDARhC,IAQ8C,8CAR9C,O;K;oCAAA,Y;MAAA,c;MAQgC,sD;MAAc,yD;MAR9C,a;K;kCAAA,iB;MA  
AA,4IAQgC,sCARhC,IAQ8C,4CAR9C,I;K;iGAUA,yB;MAAA,yC;MAgBA,8C;MAhBA,wB;QA6BI,WAAW,8B;  
QACX,aAjB8C,KAiBjC,E;QAJBb,OAKBO,oBAAW,MAAX,EAAMb,IAAK,aAAxB,C;O;KA/BX,C;mGAgBA,yB;  
MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAMb,IAAK,aAAx  
B,C;O;KafX,C;IzJZA,2E;MASI,sC;MAAA,4C;K;IATJ,mGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,4DAaQ,kB;MA  
CI,wBAAW,MAAX,C;K;IAdZ,wF;I0JewC,sC;MACpC,0B;K;;IAGJ,kC;MAUI,OAA2C,CAA3C,2BAA6B,uBAA7  
B,EAAoC,KAApC,CAA2C,e;K;IAE/C,8B;K;kDAuBI,4B;MASI,MAAM,qCAA8B,8CAA9B,C;K;;IAa4B,8C;MA  
GtC,6B;MAEmD,UAMX,M;MAPxC,kBACmD,mE;MAEnD,eAC0B,K;MAE1B,cACwC,kE;MAExC,gBACmC,g  
B;K;iGAG/B,Y;MAAQ,0C;K;0DAEZ,kB;MACI,cAAY,I;MACZ,gBAAc,M;K;IAGsE,iG;MAAA,uB;QAExE,Q;Q  
AAZ,qCAAY,8D;QACZ,sCAAA,a;QAFb,OAGA,yB;O;K;2DAJJ,+B;MAAkD,OAAc,wDAAtC,c;K;IAOyE,uH;M  
AAA,uB;QAExG,Q;QAaf,iBAae,8F;QACf,eAAK,2B;QAA6B,mC;QtMjGtB,gBAAT,Q;QsMsG0D,kB;QAJzD,sB  
AAsB,SAAK,W;QAC3B,IAAI,eAAa,eAAjB,C;UAEL,iC;UACA,mBAAY,oCAAwB,eAAxB,EAAYC,kEAAzC,C;;  
UAGZ,mBAAY,kE;;QAEhB,oBAaa,e;QAZjB,OAcA,yB;O;K;6DAfJ,0C;MAAqF,OAAc,qEAAc,c;K;IAqBzB,  
mI;MAAA,qB;QACxD,yCAAgB,uB;QAGhB,qCAAY,Y;QACZ,uCAAc,E;QACIB,W;O;K;iEATA,iC;MAGwB,w  
CAAA,mCAAb,EAAoC,kFAApC,C;K;mDAQxB,Y;MAMuB,UADC,MACD,EAIH,MAJG,EAaK,M;MAjBxB,OA  
AO,IAAP,C;QAEI,aAAa,IAAK,S;QACF,SAAL,IAAK,O;QAAL,mB;UACyB,gBAArB,0D;U3JxBhB,U;UADP,yB  
;U2JyBe,O3JxBR,sF;;Q2JuBC,WAAW,M;QAGX,IAAI,mDAAoB,MAApB,QAaj,C;;YAliB,SAAT,ezJxJV,CyJwJ  
uD,IzJxJvD,EyJwJ6D,YzJxJ7D,EyJwJoE,IzJxJpE,EAA8C,KAA9C,C;;YyJyJQ,gC;cACE,I1JzJhB,oBDgDQ,WAA  
O,c2JyG0B,C3JzG1B,CAAP,CChDR,C;c0J0JgB,Q;;cALI,O;;UAAR,c;UAQA,IAAI,MAAM,yBAAV,C;YACI,I1Jv  
KhB,oBDgDQ,W2JuHoB,0E3JvHpB,CChDR,C;;U0J0KY,gBAAc,gB;UACd,IAAK,oBAAW,MAAX,C;;K;;0EC1  
MrB,4B;MAoKI,QAhKK,SAGKG,GAhKoB,KAgKpB,I;MACR,IAAI,CAjKC,SAiKD,GAjKwB,KAiKxB,IAAiB,C  
AAjB,IAAsB,eAjKE,KAiKF,MAjKrB,SAiKL,C;QAA6C,a;;MAjK7C,OAKKO,C;K;kEAhKX,yB;MAAA,0B;MAA  
A,mC;QA2KI,QAnKK,SAmKG,GAnKe,K;QAAvB,OAAgC,OAoKzB,KApKgB,KAOkX,GAAW,CAAC,CAAC,I  
ApKF,KAOkC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApKyB,C;O;KARpC,C;4  
EAUA,4B;MAoJI,QAhJK,SAGJG,GAhJoB,KAgJpB,I;MACR,IAAI,CAjJC,SAiJD,GAjJwB,KAiJxB,IAAiB,CAAj  
B,IAAsB,eAjJE,KAiJF,MAjJrB,SAiJL,C;QAA6C,a;;MAjJ7C,OAKJO,C;K;kEAhJX,yB;MAAA,4B;MAAA,mC;QA  
2JI,QAnJK,SAmJG,GAnJe,K;QAAvB,OAAgC,QAoJzB,KApJgB,KAOjX,GAAW,CAAC,CAAC,IApJF,KAOjC,K  
AAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApJyB,C;O;KARpC,C;4EAUA,4B;MAoII,  
QAhIK,SAGIG,GAhIc,KAgId,I;MACR,IAAI,CAjIC,SAiID,GAjIkB,KAiIlB,IAAiB,CAAjB,IAAsB,eAjJI,KAiII,M  
AjIrb,SAiIL,C;QAA6C,a;;MAjI7C,OAKIO,C;K;kEAhIX,4B;MA2II,QAnIK,SAmIG,GAnIS,K;MAAjB,OAoIO,KA  
pIU,KAOIL,GAAW,CAAC,CAAC,IApIR,KAOIO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,  
EAAID,K;K;4EAIIX,yB;MAqMA,0B;MArMA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAqML,QAAQ,gBArMe,K  
AqMf,C;QACR,IAAI,gBAtMmB,KAsMnB,eAAiB,CAAjB,IAAsB,mBAtMH,KAsMG,GAAa,WAAb,CAA1B,C;U  
AA6C,W;;QAtM7C,OAuMO,C;O;KA3MX,C;kEAMA,4B;MAGNI,QAxMK,oBAAL,SAAK,CAwMG,QAxMU,K  
AwMV,C;MAxMR,OAyMO,MAzMW,KAyMN,KAAa,MAzMP,KAyMO,CAAD,KAAmB,KAAm,CAAD,aAAL,  
CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;4EAvmX,4B;MAoGI,QAhGK,SAGGG,GAhGoB,KAgGpB,I;  
MACR,IAAI,CAjGC,SAiGD,GAjGwB,KAiGxB,IAAiB,CAAjB,IAAsB,eAjGE,KAiGF,MAjGrB,SAiGL,C;QAA6  
C,a;;MAjG7C,OAKGO,C;K;kEAhGX,yB;MAAA,0B;MAAA,mC;QA2GI,QAnGK,SAmGG,GAnGe,K;QAAvB,O  
AAgC,OAoGzB,KApGgB,KAOgX,GAAW,CAAC,CAAC,IApGF,KAOGC,KAAmB,KAAK,CAAC,CAAD,IAAL,  
CAAnB,CAAD,KAAkC,EAAID,KApGyB,C;O;KARpC,C;4EAUA,4B;MAoFI,QAhFK,SAGFG,GAhFoB,KAgFpB  
,I;MACR,IAAI,CAjFC,SAiFD,GAjFwB,KAiFxB,IAAiB,CAAjB,IAAsB,eAjFE,KAiFF,MAjFrB,SAiFL,C;QAA6C,

a;;MAjF7C,OAKFO,C;K;kEAhFX,yB;MAAA,4B;MAAA,mC;QA2FI,QAnFK,SAmFG,GAnFe,K;QAAvB,OAAgC ,QAOvzB,KApFgB,KAOFX,GAAW,CAAC,CAAC,IAPFF,KAOFC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB, CAAD,KAAkC,EAAID,KApFyB,C;O;KARpC,C;4EAUA,4B;MAoEI,QAhEK,SAGEG,GAhEc,KAGEd,I;MACR,IAAI,CAjEC,SAiED,GAjEkB,KAIeIB,IAAiB,CAAjB,IAAsB,eAjEJ,KAIeI,MAjErB,SAiEL,C;QAA6C,a;;MAjE7C ,OAKEO,C;K;kEAhEX,4B;MA2EI,QAnEK,SAmEG,GAnES,K;MAAjB,OAoEO,KApEU,KAOEL,GAAW,CAAC, CAAC,IAPER,KAOEO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAIE,X,yB; MAqIA,0B;MArIA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAqIL,QAAQ,gBArIe,KAQIf,C;QACR,IAAI,gBAtImB ,KAsInB,eAAiB,CAAjB,IAAsB,mBAIH,KASIG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAtI7C,OAUIO,C;O;KA3I X,C;kEAMA,4B;MAGJI,QAxIK,oBAAL,SAAK,CAwIG,QAxIU,KAwIV,C;MAxIR,OAYIO,MAzIW,KAYIN,KAA a,MAZIP,KAYIO,CAAD,KAAmB,KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;2EA vIX,4B;MAoCI,QAhCA,SAGCQ,GAhCY,KAGCZ,I;MACR,IAAI,CAjCJ,SAiCI,GAjCgB,KAIChB,IAAiB,CAAjB, IAAsB,eAjCN,KAIcM,MAjC1B,SAiCA,C;QAA6C,a;;MAjC7C,OAKCO,C;K;iEAhCX,yB;MAAA,0B;MAAA,mC ;QA2CI,QAnCA,SAmCQ,GAnCO,K;QAAf,OAAwB,OAoCjB,KApCQ,KAOCH,GAAW,CAAC,CAAC,IAPCV,K AOCS,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApCiB,C;O;KAR5B,C;4EAUA,4 B;MAoBI,QAhBA,SAGBQ,GAhBY,KAGBZ,I;MACR,IAAI,CAjBJ,SAiBI,GAjBgB,KAIbH,IAAiB,CAAjB,IAAs B,eAjBN,KAIbM,MAjB1B,SAiBA,C;QAA6C,a;;MAjB7C,OAKBO,C;K;mEAhBX,yB;MAAA,4B;MAAA,mC;QA 2BI,QAnBA,SAmBQ,GAnBO,K;QAAf,OAAwB,QAoBjB,KApBQ,KAOBH,GAAW,CAAC,CAAC,IAPBV,KAOB S,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApBiB,C;O;KAR5B,C;4EAUA,4B;M AII,QAAQ,YAAO,KAAP,I;MACR,IAAI,aAAS,KAAT,IAAiB,CAAjB,IAAsB,eAAI,KAAJ,MAAa,SAAvC,C;QA A6C,a;;MAC7C,OAAO,C;K;mEAGX,4B;MAQI,QAAQ,YAAO,K;MACf,OAAO,KAAK,QAAW,CAAC,CAAC,I AAM,KAAP,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAGX,yB;MAGEA,0B ;MAhEA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAGEL,QAAQ,gBAhEe,KAGef,C;QACR,IAAI,gBAjEmB,KAIEn B,eAAiB,CAAjB,IAAsB,mBAjEH,KAIeG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAJE7C,OAKEO,C;O;KATEx,C; kEAMA,4B;MA2EI,QAnEK,oBAAL,SAAK,CAmEG,QAnEU,KAmEV,C;MAnER,OAoEO,MApEW,KAOEN,KA Aa,MApEP,KAOEO,CAAD,KAAmB,KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;6 EAIE,X,yB;MAGDA,0B;MAhDA,mC;QAIS,cAAe,oBAAN,KAAm,C;QAGDpB,QAhDA,SAGDQ,KAAO,OAAP,C; QACR,IAjDA,SAiDI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjD1B,SAiD0B,CAA1B,C;UAA6C, W;;QAJD7C,OAKDO,C;O;KATDX,C;mEAMA,yB;MAAA,0B;MAAA,mC;QAQS,cAAU,oBAAN,KAAm,C;QAm Df,QAnDA,SAmDQ,QAAO,OAAP,C;QAnDR,OAAYB,OAoDIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB, KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAPDkB,S;O;KAR7B,C;6EAUA,yB;MAGC A,0B;MAhCA,mC;QAIS,cAAe,oBAAN,KAAm,C;QAGCpB,QAhCA,SAGCQ,KAAO,OAAP,C;QACR,IAjCA,SAi CI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjC1B,SAiC0B,CAA1B,C;UAA6C,W;;QAJC7C,OAKC O,C;O;KATCX,C;mEAMA,yB;MAAA,4B;MAAA,mC;QAQS,cAAU,oBAAN,KAAm,C;QAmCf,QAnCA,SAmCQ ,QAAO,OAAP,C;QAnCR,OAAYB,QAoCIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAm,CAAD,aAA L,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAPCkB,S;O;KAR7B,C;6EAUA,yB;MAGBA,0B;MAhBA,mC; QAIS,cAAe,oBAAN,KAAm,C;QAGBpB,QAhBA,SAGBQ,KAAO,OAAP,C;QACR,IAjBA,SAiBI,KAAS,OAAT,e AAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjB1B,SAiB0B,CAA1B,C;UAA6C,W;;QAJB7C,OAKBO,C;O;KATBX,C;m EAMA,4B;MAQS,cAAU,oBAAN,KAAm,C;MAMbF,QAnBA,SAmBQ,QAAO,OAAP,C;MAnBR,OAoBO,MAAK ,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CA pBkB,Q;K;6EAE7B,yB;MAAA,0B;MAAA,mC;QAII,QAAQ,cAAO,KAAP,C;QACR,IAAI,cAAS,KAAT,eAAiB, CAAjB,IAAsB,mBAAI,KAAJ,GAAa,SAAb,CAA1B,C;UAA6C,W;;QAC7C,OAAO,C;O;KANX,C;mEASA,4B;M AQI,QAAQ,iBAAO,KAAP,C;MACR,OAAO,MAAK,UAAa,MAAM,KAAm,CAAD,KAAmB,KAAm,CAAD,aAA L,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;kEAGX,yB;MpGiqB2C,iB;MoGjqB3C,mC;QAUI,QAAQ,Y AAO,K;QACJ,iBAAS,G;QAAT,S;UAAAsB,OpGspBc,MAAiC,MoGtpB/C,CpGspB+C,CoGtpB/C,KpGspBc,MAAi C,MoGtpBrC,KpGspBqC,C;;QoGtpBhF,OAAO,OAAGD,IAAI,KAApD,GAA+D,C;O;KAXIE,C;mEAca,yB;MpG 0I6C,iB;MoGI17C,mC;QAKCI,QAxBK,SAwBG,GAXBY,K;QAYBT,iBAAK,G;QAAL,S;UAAy,OpGuG0B,MAA W,MoGvGrC,CpGuGqC,CoGvGrC,KpGuG0B,MAAW,MoGhIxC,KpGgIwC,C;;QoGhI5D,OAYBO,OAAsC,IAzBz B,KAYbB,GAAqD,C;O;KANChE,C;mEAYA,yB;MpG8H6C,iB;MoG9H7C,mC;QASBI,QAZA,SAYQ,GAZO,K;Q

AaJ,iBAAK,G;QAAL,S;UAA Y,OpGuG0B,MAAW,MoGvGrC,CpGuGqC,CoGvGrC,KpGuG0B,MAAW,MoGpH7  
C,KpGoH6C,C;;QoGpH5D,OAAO,OAA sC,IAb9B,KAAr,GAAqD,C;O;KAvBhE,C;mEAYA,yB;MpGkH6C,iB;Mo  
GIH7C,mC;QAUI,QAAQ,YAAO,K;QACJ,iBAAK,G;QAAL,S;UAA Y,OpGuG0B,MAAW,MoGvGrC,CpGuGqC,  
CoGvGrC,KpGuG0B,MAAW,MoGvG3B,KpGuG2B,C;;QoGvG5D,OAAO,OAA sC,IAAI,KAA1C,GAAqD,C;O;K  
AXhE,C;4ECnTA,yB;MAAA,8B;MAAA,4B;QAOyC,Q;QAAA,gFAAoB,C;O;KAP7D,C;ICM0B,4C;MA+CtB,qC;  
MA/CuB,kB;MAAgB,kB;MAAgB,kB;MAMvD,iBAAsB,iBAAU,UAAV,EAAiB,UAAjB,EAAwB,UAAxB,C;K;0  
CAEtB,+B;M5MWA,IAAI,E4MViB,CAAT,sBAAY,GAAZ,KAA4C,CAAT,sBAAY,GAA/C,MAA+E,CAAT,sBA  
AY,GAAIF,C5MUR,CAAJ,C;QACI,c4MVI,2E;Q5MWJ,MAAM,gCAAyB,OAAQ,WAAjC,C;;M4MTN,OAAO,C  
AAA,KAAM,IAAI,EAAV,KAAgB,KAAM,IAAI,CAA1B,IAA+B,KAA/B,I;K;uCAGX,Y;MAGkC,OAAE,UAAF,  
oBAAS,UAA T,SAAGB,U;K;qCAEID,iB;MAEwB,gB;MADpB,IAAI,SAAS,KAAb,C;QAAoB,OAAO,I;MACP,iE;  
MAAD,mB;QAA6B,OAAO,K;;MAAvD,mBAAmB,M;MACnB,OAAO,IAAK,UAA L,KAAgB,YAAa,U;K;uCAGx  
C,Y;MAA+B,qB;K;8CAE/B,iB;MAAoD,wBAAU,KAAM,UAAhB,I;K;gDAEpD,wB;MAKI,OAAA,IAAK,MAAL  
,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAd,IACf,IAAK,MAAL,IAAc,KADtB,C;K;gDAGJ,+B;MAKI,OA  
AA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAd,KACd,IAAK,MAAL,GAAa,KAAb,KAA sB,  
IAAK,MAAL,KAAc,KAAd,IACf,IAAK,MAAL,IAAc,KADrB,CADc,CAA vB,C;K;IAIJ,mC;MAAA,uC;MACI,2B  
AluC,G;MAEvC,eAIoC,uCAA0B,M;K;;;IAXIE,+C;MAAA,8C;QAAA,6B;;MAAA,uC;K;;IA9CA,iD;MAAA,uD;  
MAG6C,0BAAK,KAAL,EAA Y,KAAZ,EAAMB,CAAnB,C;MAH7C,Y;K;IA6DJ,qC;MAAA,yC;K;8CAEI,Y;MA  
C2B,yBAAc,CAAd,EAAiB,CAAjB,EAAoB,EAApB,C;K;;IAH/B,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;4FCx  
DI,yB;MAAA,2D;MAAA,4B;QAAQ,MAAM,6BAAoB,6BAApB,C;O;KAAd,C;;;ICSJ,uB;MAG2C,+BAAoB,KA  
ApB,C;K;4EAE3C,wC;MAO4F,sB;K;IAE5F,6C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,2C;MAAA,8C;O;MAK  
I,wF;MAKA,sF;MAMA,wE;K;;IAXA,yD;MAAA,iC;MAAA,iD;K;;IAKA,wD;MAAA,iC;MAAA,gD;K;;IAMA,iD  
;MAAA,iC;MAAA,yC;K;;IAhBJ,uC;MAAA,iJ;K;;IAAA,4C;MAAA,a;aAAA,c;UAAA,sD;aAAA,a;UAAA,qD;aA  
AA,M;UAAA,8C;;UAAA,gE;;K;;IAyBA,+B;MAAA,mC;K;;;IAAA,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAG  
oC,qC;MACHc,qBAAsC,W;MACtC,gBAA2B,iC;K;uFAGvB,Y;MAMW,Q;MALP,IAAI,kBAAW,iCAAf,C;QACI  
,gBAAS,mC;QACT,qBAAc,I;;MAGIB,OAAO,gF;K;6CAGf,Y;MAAwC,yBAAW,iC;K;wCAEnD,Y;MAAkC,OA  
AI,oBAAJ,GAA2B,SAAN,UAA M,CAA3B,GAA2C,iC;K;8CAE7E,Y;MAAkC,+BAAoB,UAApB,C;K;;IAGG,oC;  
MAAC,4B;K;wEAAA,Y;MAAA,2B;K;kDAEtC,Y;MAAwC,W;K;6CAExC,Y;MAAkC,OAAM,SAAN,UAA M,C;  
K;;oFC2C5C,yB;MAAA,gD;MAAA,4B;QAM6C,OAAMB,aAAIB,YAA Y,GAAM,C;O;KANhE,C;oGAQA,yB;Mx  
G7FA,iB;MwG6FA,4B;QAMqD,OxG7FM,MAAO,OwG6FZ,YAA Y,GxG7FA,CwG6Fb,GAA6C,EAA7C,I;O;KA  
NrD,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAMB,sBAAlB,YAAW,GAAO,C;O;KANzE,C;8FAQA,yB;  
MAAA,0D;MAAA,0B;MAAA,4B;QAOmD,OA AuC,OA ApB,kBAAlB,YAA Y,GAAM,CAAoB,C;O;KAP1F,C;4F  
ASA,yB;MAAA,wD;MAAA,0B;MAAA,4B;QAOkD,OAA2B,OAAnB,iBAAR,SAAQ,CAAMB,C;O;KAP7E,C;IA  
UA,2C;MAaI,OAA+E,OAA9E,SAAQ,KAAI,WAAa,CAAjB,CAAR,GAAkD,CAAIB,YAA Y,GAAM,MAAK,CA  
AL,IAAU,WAAa,CAA vB,CAA4B,C;K;IAEnF,4C;MAaI,OAA+E,OAA9E,SAAQ,IAAI,CAAJ,IAAS,WAAa,CAAt  
B,CAAR,GAAwD,CAAIB,YAA Y,GAAM,OA AK,WAAa,CAAIB,CAAsB,C;K;oFAEnF,yB;MAAA,gD;MAAA,4  
B;QAM8C,OAAqB,aAApB,YAA Y,KAAQ,C;O;KANnE,C;oGAQA,yB;MxGtKA,iB;MwGsKA,4B;QAOI,OxGvK  
uD,MAAO,OwGuK7D,YAA Y,KxGvKiD,CwGuK9D,GAA+C,EAA/C,I;O;KAPJ,C;sGASA,yB;MAAA,kE;MAAA  
,4B;QAMuD,OAAqB,sBAApB,YAAW,KAAS,C;O;KAN5E,C;8FAQA,yB;MAAA,0D;MAAA,4B;MAAA,4B;QA  
OqD,OAAyC,QAApB,kBAApB,YAA Y,KAAQ,CAAoB,C;O;KAP9F,C;4FASA,yB;MAAA,wD;MAAA,4B;MAA  
A,4B;QAOoD,OAA2B,QAAnB,iBAAR,SAAQ,CAAMB,C;O;KAP/E,C;IAUA,2C;MAaI,OAAoF,QAAnF,SAAQ,  
KAAI,WAAa,EAAjB,CAAR,GAAqD,CAApB,YAA Y,KAAQ,MAAK,EAAL,IAAW,WAAa,EAAxB,CAA8B,C;K;  
IAExF,4C;MAaI,OAAoF,QAAnF,SAAQ,IAAI,EAAJ,IAAU,WAAa,EAA vB,CAAR,GAA4D,CAApB,YAA Y,KA  
AQ,OA AK,WAAa,EAAIB,CA AuB,C;K;0E/MIRxF,yB;MAaA,kF;MAbA,wB;QAU BI,IAAI,CAB I,KAAr,C;UACI,c  
Ada,qB;UAeb,MAAM,8BAAYB,OAAQ,WAAjC,C;;O;KAZbd,C;0EAaA,yB;MAAA,kF;MAAA,qC;QAUI,IAAI,C  
AAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,8BAAYB,OAAQ,WAAjC,C;;O;KAZd,C;sFAGBA,yB;MAWA,kF;M  
AXA,wB;QAQW,yB;QAeP,IAfsB,KAelB,QA AJ,C;UACI,cAhB2B,0B;UAiB3B,MAAM,8BAAYB,OAAQ,WAAjC  
,C;;UAEN,wBAnBkB,K;;QAAtB,4B;O;KARJ,C;wFAWA,yB;MAAA,kF;MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,c  
AAc,a;UACd,MAAM,8BAAYB,OAAQ,WAAjC,C;;UAEN,OAAO,K;;O;KAhBf,C;oEAoBA,yB;MAaA,4E;MAbA,

wB;QAuBI,IAAI,CABe,KAaN,C;UACI,cAdW,e;UAeX,MAAM,2BAAsB,OAAQ,WAA9B,C;;O;KAZBd,C;sEAaA ,yB;MAAA,4E;MAAA,qC;QAUI,IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B, C;;O;KAZd,C;kFAgBA,yB;MAcA,4E;MAdA,wB;QAWW,uB;QAeP,IAfoB,KAehB,QAAl,C;UACI,cAhByB,0B;U AiBzB,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;QAAPB,0B;O;KAXJ,C;oFAcA,yB;MAAA,4E; MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,OAAO,K;;O; KAhBf,C;oEAqBA,yB;MAAA,4E;MAAA,0B;QAMiD,MAAM,2BAAsB,OAAQ,WAA9B,C;O;KANvD,C;I8CnHi C,uB;MA2D7B,8B;MA1DA,kB;K;mFAS8B,Y;MAAQ,iD;K;mFAMR,Y;MAAQ,gD;K;wFAItC,yB;MAAA,gB;M AAA,8B;MAAA,mB;QAWgB,Q;QADR,mB;UADJ,OACiB,I;;UADjB,OAEY,2E;O;KAXhB,C;uCACa,Y;MAQQ, kBADE,UACF,kB;QADJ,OACkB,UAAM,U;;QADxB,OAEY,I;K;gCAGhB,Y;MAOQ,kBADE,UACF,kB;QADJ,O ACkB,UAAM,W;;QADxB,OAEY,sBAAU,UAAV,O;K;IAKhB,4B;MAAA,gC;K;wHAKI,yB;MAAA,iC;MAAA,w B;QAOI,uBAAO,KAAP,C;O;KAPJ,C;wHASA,yB;MAAA,kD;MAAA,iC;MAAA,4B;QAOI,uBAAO,cAAc,SAAd, CAAP,C;O;KAPJ,C;;;IADJ,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;IAwBsB,mC;MAClB,0B;K;sCAGA,iB;MAA 4C,+CAAoB,uBAAa,KAAM,UAAAnB,C;K;wCACHe,Y;MAA+B,OAAU,SAAV,cAAU,C;K;wCACzC,Y;MAAkC, oBAAU,cAAV,M;K;;;gCA/F1C,Y;MAAA,c;MAOI,sD;MAPJ,a;K;8BAAA,iB;MAAA,2IAOI,sCAPJ,G;K;IAmG A,kC;MAOI,OAAO,mBAAQ,SAAR,C;K;IAEX,mC;MAQI,IAAI,8CAAJ,C;QAA6B,MAAM,eAAM,U;K;gFAG7C ,yB;MAAA,4B;MAAA,qB;MAxCQ,kD;MAwCR,wB;QAOW,Q;;UACI,OAIDH,WakDW,OAIDX,C;;UAmDN,gC ;YACS,OA3CH,WAAO,cA2CI,CA3CJ,CAAP,C;;YAwCD,O;;QAAP,W;O;KAPJ,C;kFAcA,yB;MAAA,4B;MAAA ,qB;MAiDQ,kD;MAAsDR,mC;QAOW,Q;;UACI,OAHEH,WAgEW,gBAhEX,C;;UAiEN,gC;YACS,OAzDH,WAAO, cAyDI,CAzDJ,CAAP,C;;YAsDD,O;;QAAP,W;O;KAPJ,C;8EAgBA,yB;MAAA,oD;MAAA,gB;MAAA,8B;MAAA ,4B;QAOW,Q;QADP,yB;QACA,OAAO,gF;O;KAVX,C;+EAaA,yB;MAAA,gB;MAAA,8B;MAAA,uC;QAegB,U ADL,M;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,yF;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAdJ,C;kFAoBA,yB; MAAA,gB;MAAA,8B;MAAA,0C;QAOW,Q;QADP,IAAI,mBAAJ,C;UAAe,OAAO,Y;QAcTb,OAAO,gF;O;KAV X,C;qEAaA,yB;MAAA,gB;MAAA,8B;MAAA,kD;QAIb0B,UADf,M;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,m BAAU,gFAAV,C;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAHBJ,C;mEAwBA,yB;MAAA,4B;MAAA,gB;MAAA,8 B;MAAA,uC;YAE8C,I;YADnC,M;QACH,wB;UAAa,gB;UAAO,SA7JhB,WA6JwB,UAAU,gFAAV,CA7JxB,C;;U A8JI,oBAAO,eAAP,C;QAFZ,a;O;KAdJ,C;gFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,iC;MA1GA,qB;MAiDQ,k D;MAgKR,uC;QAWW,Q;QACH,wB;UA/GG,U;;YA+GkC,U;YA9G9B,SAhEH,gBA8KuB,UAAU,sFAAV,CA9K vB,C;;YAiEN,gC;cACS,SAzDH,gBAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UA+GU,a;;UACL,uBAAO,eAAP,C; QAFZ,W;O;KAXJ,C;wEAIbA,yB;MAAA,4B;MAAA,uC;QAcW,Q;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,gB;; UACO,OAnMX,WAmMmB,UAAU,SAAV,CAnMnB,C;;QAIrM,W;O;KAdJ,C;wFAoBA,yB;MA/IA,4B;MAAA,q B;MAiDQ,kD;MAqMR,uC;QAWW,Q;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,gB;;UApJL,U;;YACI,SAhEH,WA oNkB,oBApNIB,C;;YAiEN,gC;cACS,SAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UAqJK,a;;QAFZ,W;O; KAXJ,C;4EAmBA,6B;MAUI,Q;MAAA,iD;QAAYB,Y;;MACzB,OAAO,S;K;4EAGX,yB;MAAA,gB;MAAA,8B;M AAA,oC;QAU0B,Q;QAAtB,IAAI,mBAAJ,C;UAAe,OAAO,gFAAP,C;;QAcF,OAAO,S;O;KAXX,C;I3CtTgC,sC; MAAC,uB;QAAA,UAAkB,kC;mBAA4C,O;;K;;0DAE/F,yB;MAAA,2D;MAAA,mB;QAKoC,MAAM,8B;O;KAL1 C,C;oEAOA,yB;MAAA,2D;MAAA,yB;QAMkD,MAAM,6BAAoB,sCAAmC,MAAvD,C;O;KANxD,C;gEUA,iB ;MAUI,OAAO,O;K;kEAGX,4B;MAUI,OAAO,gB;K;oEAGX,2B;MAUI,OAAgB,MAAT,QAAS,C;K;oEAGpB,4B ;MAUI,gB;MACA,OAAO,S;K;kEAGX,4B;MAWI,MAAM,SAAN,C;MACA,OAAO,S;K;kEAGX,4B;MAUI,OAA O,MAAM,SAAN,C;K;sEAGX,gC;MAWI,OAAW,UAAU,SAAV,CAAJ,GAAqB,SAArB,GAA+B,I;K;8EAG1C,g C;MAWI,OAAW,CAAC,UAAU,SAAV,CAAL,GAAsB,SAAtB,GAAgC,I;K;wEAG3C,yB;MAWI,iBAAc,CAAd, UAAsB,KAAtB,U;QACI,OAAO,KAAP,C;;K;wE6MjJR,iB;MAIkF,Y;K;ICY9C,6B;MACHc,kB;MACA,oB;K;8B AGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,M;K;;gCAvB7C,Y;MAGBI,iB;K;gCAhBJ,Y;MAiBI,kB;K;kCAjB J,yB;MAAA,gBAgBI,qCAhBJ,EAIbI,wCAjBJ,C;K;8BAAA,Y;MAAA,c;MAGBI,sD;MACA,uD;MAjBJ,a;K;4BA AA,iB;MAAA,4IAGBI,sCAhBJ,IAiBI,wCAjBJ,I;K;IA0BA,6B;MAMoD,gBAAK,SAAL,EAAW,IAAX,C;K;IAEp D,8B;MAI8C,iBAAO,eAAP,EAAc,gBAAd,E;K;IAiBD,sC;MACzC,kB;MACA,oB;MACA,kB;K;gCAGA,Y;MAG yC,aAAG,UAAH,UAAW,WAAX,UAAoB,UAApB,M;K;;kCAxB7C,Y;MAGBI,iB;K;kCAhBJ,Y;MAiBI,kB;K;kC AjBJ,Y;MAkBI,iB;K;oCAIBJ,gC;MAAA,kBAgBI,qCAhBJ,EAIbI,wCAjBJ,EAKBI,qCAIBJ,C;K;gCAA,Y;MAA A,c;MAGBI,sD;MACA,uD;MACA,sD;MAIBJ,a;K;8BAAA,iB;MAAA,4IAGBI,sCAhBJ,IAiBI,wCAjBJ,IAKBI,sCA

IBJ,I;K;IA2BA,8B;MAImD,iBAAO,eAAP,EAAC,gBAAd,EAAsB,eAAtB,E;K;I7NIE1B,qB;MAErB,6B;MAFwD,g  
B;K;IAExD,2B;MAAA,+B;MACI,iBAGoC,UAAM,CAAN,C;MAEpC,iBAGoC,UAAM,MAAN,C;MAEpC,kBAG  
mC,C;MAEnC,iBAGkC,C;K;;;IANtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;kGAsBA,iB;MAOmE,OAAa,0BA  
2O1C,SAAL,GAAiB,GA3O8B,EAAU,KA2OpD,KAAL,GAAiB,GA3O8B,C;K;sGAehF,iB;MAM2D,OAAa,0BA  
mOIC,SAAL,GAAiB,GAnOsB,EAAU,KEoO5C,KAAL,GAAiB,KFpOsB,C;K;sGAExE,yB;MA0PA,6B;MC3PA,8  
C;MDCA,wB;QAMyD,OCAS,YAAiB,CD6PhD,cAAU,SAAL,GAAiB,GAAtB,CC7PgD,MAAjB,EDAe,KCAc,KA  
A7B,C;O;KDNIE,C;sGAQA,yB;MA4PA,WAS6D,wB;MAT7D,+B;MiB7PA,gD;MjBCA,wB;QAM0D,OiBAS,aA  
AkB,CjB+PhD,eAAW,oBAAL,SAAK,CAAL,UAAN,CiB/PgD,MAAiB,EjBAgB,KiBAc,KAA9B,C;O;KjBNnE,C;  
4FAQA,yB;MA0OA,6B;MA1OA,wB;QAEsD,OCMD,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,M  
AAK,GAAW,CD2O5C,cAjPsC,KAiP5B,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;O;KDRrD,C;4FAGA,y  
B;MAuOA,6B;MAvOA,wB;QAEuD,OCGF,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,MAAK,GA  
AW,CC4O5C,cF/OuC,KE+O7B,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;O;KDLrD,C;4FAGA,yB;MAoO  
A,6B;MApOA,wB;QAEqD,OCAA,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,MAAK,GDAI,KCAO  
,KAAZ,IAAf,C;O;KDFrD,C;4FAGA,yB;MA2OA,WAS6D,wB;MAT7D,+B;MA3OA,wB;QAEuD,OiBAA,eAAW,  
CjBkP7B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBIP6B,MAAK,KjBAI,KiBAO,KAAZ,CAAhB,C;O;KjBFvD,C;  
8FAIA,yB;MA6NA,6B;MA7NA,wB;QAEuD,OCMD,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,M  
AAK,GAAY,CD8N9C,cApOwC,KAoO9B,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KDRtD,C;8FAGA,  
yB;MA0NA,6B;MA1NA,wB;QAEwD,OCGF,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,MAAK,G  
AAY,CC+N9C,cFIOyC,KEkO/B,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KDLtD,C;8FAGA,yB;MAuN  
A,6B;MAvNA,wB;QAEsD,OCAA,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,MAAK,GDAK,KCA  
O,KAAZ,IAAf,C;O;KDFtD,C;8FAGA,yB;MA8NA,WAS6D,wB;MAT7D,+B;MA9NA,wB;QAEwD,OiBAA,eAA  
W,CjBqO9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBrO8B,MAAK,UjBAK,KiBAO,KAAZ,CAAhB,C;O;KjBFx  
D,C;8FAIA,yB;MAGNA,6B;MAhNA,wB;QAEuD,OCMD,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,C  
CjN6B,MAAK,EAAY,CDiN9C,cAvNwC,KAuN9B,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KDRtD,C  
;8FAGA,yB;MA6MA,6B;MA7MA,wB;QAEwD,OCGF,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,CCj  
N6B,MAAK,EAAY,CCKN9C,cFrNyC,KEqN/B,KAAL,GAAiB,KAAtB,CDiN8C,MAAZ,CAAf,C;O;KDLtD,C;8F  
AGA,yB;MA0MA,6B;MA1MA,wB;QAEsD,OCAA,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,CCjN6B  
,MAAK,EDAK,KCAO,KAAZ,CAAf,C;O;KDFtD,C;8FAGA,yB;MAiNA,WAS6D,wB;MAT7D,+B;MAjNA,wB;Q  
AEwD,OiBAA,eAAW,CjBwN9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBxN8B,MAAK,UjBAK,KiBAO,KAAZ  
,CAAhB,C;O;KjBFxD,C;0FAIA,yB;MAmMA,6B;MC7LA,4C;MDNA,wB;QAEqD,OCMD,WDoMjB,cAAU,SAA  
L,GAAiB,GAAtB,CCpMiB,EDoMjB,cA1MoC,KA0M1B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KDRpD,C;0FAG  
A,yB;MAGMA,6B;MC7LA,4C;MDHA,wB;QAEsD,OCGF,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,ECq  
MjB,cFxMqC,KEwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KDLpD,C;0FAGA,yB;MA6LA,6B;MC7LA,4C;M  
DAA,wB;QAEoD,OCAA,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDakB,KCAIB,C;O;KDFpD,C;0FA  
GA,yB;MAoMA,WAS6D,wB;MAT7D,+B;MiBpMA,8C;MjBAA,wB;QAEsD,OiBAA,YjB2MjB,eAAW,oBAAL,S  
AAK,CAAL,UAAN,CiB3MiB,EjBAmb,KiBAnB,C;O;KjBFtD,C;0FAIA,yB;MA5LA,6B;MCxKA,kD;MDdA,wB;  
QAMqD,OCcD,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,cAzLoC,KAYL1B,KAAL,GAAiB,GAAt  
B,CC3KiB,C;O;KDPpD,C;0FAOA,yB;MA+KA,6B;MCxKA,kD;MDPA,wB;QAMsD,OCOF,cD2KjB,cAAU,SA  
AL,GAAiB,GAAtB,CC3KiB,EC4KjB,cFnLqC,KEmL3B,KAAL,GAAiB,KAAtB,CD5KiB,C;O;KDbpD,C;0FAOA,  
yB;MAwKA,6B;MCxKA,kD;MDAA,wB;QAMoD,OCAA,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDAK  
B,KCAIB,C;O;KDNpD,C;0FAOA,yB;MA2KA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MjBAA,wB;QAMsD,OiB  
AA,ejB8KjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CiB9KiB,EjBAmb,KiBAnB,C;O;KjBNtD,C;oGAQA,yB;MAy  
JA,6B;MC7LA,4C;MDoCA,wB;QAMiD,OCxCG,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDoMjB,cA5  
JqC,KA4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KDKcP,D,C;oGAOA,yB;MAKJA,6B;MC7LA,4C;MD2CA,wB;  
QAMkD,OC/CE,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,ECqMjB,cFtJsC,KEsJ5B,KAAL,GAAiB,KAA  
tB,CDrMiB,C;O;KDYcP,D,C;oGAOA,yB;MA2IA,6B;MC7LA,4C;MDkDA,wB;QAMgD,OCtDI,WDoMjB,cAAU,  
SAAL,GAAiB,GAAtB,CCpMiB,EDsDmB,KCtDnB,C;O;KDGdP,D,C;oGAOA,yB;MA8IA,WAS6D,wB;MAT7D,+  
B;MiBpMA,8C;MjBsDA,wB;QAMkD,OiB1DI,YjB2MjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CiB3MiB,EjB0D

oB,KiB1DpB,C;O;KjBoDtD,C;0FAQA,yB;MA4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MA3LI,wB;QAQ6C,OA8LR,eAAW,OC5OI,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,cA7H4B,KA6HIB,KAAL,GAAiB,GAAtB,CC3KiB,CAkLf,KD0DW,CAAX,C;O;KATMrC,C;0FASA,yB;MAMHA,6B;MCxKA,kD;MCwOJ,4B;MAAA,iC;MFnLI,wB;QAQ+C,OEslR,gBAAy,QD7OC,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EC4KjB,cFrH8B,KEqHpB,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KF9LvC,C;0FASA,yB;MA0GA,6B;MCxKA,kD;MD8DA,wB;QAQ2C,OCHEs,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDgES,KChET,C;O;KDwDpD,C;0FASA,yB;MA2GA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MjBgEA,wB;QAQ6C,OiBIES,ejB8KjB,eAAW,oBAAL,SAAK,CAAL,UAN,CiB9KiB,EjBkEU,KiBIEV,C;O;KjB0DtD,C;0EAUA,yB;MAAA,0B;MAAA,+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,0B;MAAA,+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,8C;MAuEA,6B;MAvEA,wB;QAE8D,0BA8E3B,cAAU,SAAL,GAAiB,GAAtB,CA9E2B,EA8E3B,cA9EoD,KA8E1C,KAAL,GAAiB,GAAtB,CA9E2B,C;O;KAF9D,C;0FAIA,yB;MAAA,+B;M6LxOJ,0B;M7LwOI,wB;QAEEmD,sB6LvOgC,O7LuO1B,IAAK,K6LvOX,G7LuOoB,KAAM,K6LvOM,C7LuOhC,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;M6LTOJ,0B;M7LsOI,wB;QAEkD,sB6LrO+B,O7LqOzB,IAAK,K6LrOX,G7LqOmB,KAAM,K6LrOM,C7LqO/B,C;O;KAFID,C;0FAGA,yB;MAAA,+B;M6LpOJ,0B;M7LoOI,wB;QAEEmD,sB6LnOgC,O7LmO1B,IAAK,K6LnOX,G7LmOoB,KAAM,K6LnOM,C7LmOhC,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;M6LlOJ,0B;M7LkOI,mB;QAEiC,sB6LjOqB,OAAP,C7LlOR,S6LjOe,C7LiOrB,C;O;KAFjC,C;gFAIA,Y;MASmC,gB;K;kFACnC,yB;M6L1OJ,4B;M7L0OI,mB;QASqC,O6LhPiD,Q7LgP5C,S6LhPY,G7LgPE,G6LhP8B,C;O;K7LuOtF,C;8EAUA,Y;MASiC,OAAL,SAAL,GAAiB,G;K;gFACID,yB;MAAA,WASqD,wB;MATrD,mB;QASmC,OAAL,SAAK,CAAL,U;O;KATnC,C;kFAWA,Y;MAEqC,W;K;0FACrC,yB;MAAA,iC;M6L5QJ,4B;M7L4QI,mB;QASuC,uB6LIR+C,Q7LkRnC,S6LIRG,G7LkRW,G6LIRqB,C7LkR/C,C;O;KATvC,C;gFAUA,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,GAAtB,C;O;KATnC,C;kFAUA,yB;MAAA,WAS6D,wB;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,UAN,C;O;KATrC,C;kFAWA,Y;MAMqC,OAAPDC,SAAL,GAAiB,G;K;0FAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,G;K;+BA6DID,Y;MAAyC,OAQ,CA7DX,SAAL,GAAiB,GA6DD,Y;K;+BA1UrD,Y;MAAA,c;MAG4D,qD;MAH5D,a;K;6BAAA,iB;MAAA,2IAG4D,oCAH5D,G;K;wEA8UA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;0EAWA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW2C,sBAAW,OAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAWyC,sBAAW,OAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW0C,sBAAW,OAAL,SAAK,SAAX,C;O;KAX1C,C;Igc9WA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OhCyUP,UgCzUO,AAQ,KAAR,ChCyUP,C;K;uCgCvUtC,wB;MAOI,aAAQ,KAAR,IAAiB,KhCiOc,K;K;kFgC7NL,Y;MAAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAGxD,oC;MAAiC,wB;MAAhC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;8CACvC,Y;MAAyD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAlB,C;QAAA,OhCmTO,UgCnTiB,aAAM,mBAAN,EAAM,2BAAN,OhCmTjB,C;QgCnT+C,MAAM,2BAAuB,YAAM,WAA7B,C;K;0CAG3F,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAlO,K;MAExC,OAAlE,WAAR,YAAQ,EAAS,OhC2MO,KgC3MhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;Qd0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;QACrB,6B;QAAhB,OAAGB,gBAAhB,C;UAGB,2B;Uc1nD6B,2Bd0nDR,Oc1nDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,Ed0nD/B,OIBn7CF,KgCvMiC,C;Ud0nD9C,IAAI,OAAlC;YAAyB,aAAO,K;YAAP,e;QAC/C,aAAO,I;Mc3nDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;IA/CvD,sC;MAAA,oD;MACgC,uBAAK,cAAU,IAAV,CAAL,C;MADhC,Y;K;+oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAyDA,yB;MAAA,yC;MAWsC,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,ChCsLV,K;S;O;MgCjMvC,6B;QAWI,OAAlO,oBAAW,+BAAU,IAAV,GAAGB,uBAAhB,CAAX,C;O;KAXX,C;kFACa,oB;MAGqE,e;K;I/BtE7C,oB;MAEpB,4B;MAFuD,gB;K;IAEvD,0B;MAAA,8B;MACI,iBAGmC,SAAK,CAAL,C;MAEnC,iBAGmC,SAAK,EAAL,C;MAEnC,kBAGmC,C;MAEnC,iBAGkC,E;K;+IANBtC,sC;MAAA,qC;QAAA,oB;MAAA,8B;K;oGAsBA,yB;MD2QA,6B;MC3PA,8C;MAhBA,wB;QAM0D,OAiBQ,YAAy,IAAK,KAAjB,EAA6B,CD6P5D,cC9QsC,KD8Q5B,KAAL,GAAiB,GAAtB,CC7P4D,MAA7B,C;O;KAvBIE,C;oGAQA,yB;MCoQA,6B;MD5PA,8C;MARA,wB;QAM2D,OASO,YAAy,IAAK,KAAjB,EAA6B,CC8P5D,cDvQuC,KCuQ7B,KAAL,GAAiB,KAAtB,CD9P4D,MAA7B,C;O;KAFIE,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAOKE,mBAAy,IAAK,KAAjB,EAAuB,KAAM,KAA7B,C;O;KAPIE,C;oGASA,yB;MAGRA,kB



AS6D,sB;MAT7D,+B;MgBjRA,gD;MhBCA,wB;QAM0D,OgBAS,aAAkB,ChBmRhD,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBnRgD,MAAIB,EhBAGB,KgBAC,KAA9B,C;O;KhBNnE,C;0FAQA,yB;MD00A,6B;MC10A,wB;QAEsD,OAMD,cAAK,IAAK,KAAK,GAAW,CD205C,cCjP6B,KDiPnB,KAAL,GAAiB,GAAtB,CC304C,MAAX,IAAf,C;O;KARrD,C;0FAGA,yB;MCwOA,6B;MDxOA,wB;QAEuD,OAGF,cAAK,IAAK,KAAK,GAAW,CC405C,cD/08B,KC+OpB,KAAL,GAAiB,KAAtB,CD504C,MAAX,IAAf,C;O;KALrD,C;0FAGA,yB;MAAA,6B;MAAA,wB;QAEqD,qBAAK,IAAK,KAAK,GAAM,KAAX,IAAf,C;O;KAFrD,C;0FAGA,yB;MA+PA,kBAS6D,sB;MAT7D,+B;MA/PA,wB;QAEuD,OgBAA,eAAW,ChBsQ7B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBtQ6B,MAAK,KhBAI,KgBAO,KAAZ,CAAHb,C;O;KhBFvD,C;4FAIA,yB;MD6NA,6B;MC7NA,wB;QAEuD,OAMD,cAAK,IAAK,KAAK,GAAY,CD8N9C,cCpO+B,KDoOrB,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KARtD,C;4FAGA,yB;MC2NA,6B;MD3NA,wB;QAEwD,OAGF,cAAK,IAAK,KAAK,GAAY,CC+N9C,cDIOgC,KCkOtB,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,qBAAK,IAAK,KAAK,GAAM,KAAM,KAAX,IAAf,C;O;KAFtD,C;4FAGA,yB;MAkPA,kBAS6D,sB;MAT7D,+B;MAIPA,wB;QAEwD,OgBAA,eAAW,ChByP9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBzP8B,MAAK,UhBAK,KgBAO,KAAZ,CAAHb,C;O;KhBFxD,C;4FAIA,yB;MDgNA,6B;MChNA,wB;QAEuD,OAMD,cAAe,YAAV,IAAK,KAAK,EAAY,CDiN9C,cCvN+B,KDuNrB,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KARtD,C;4FAGA,yB;MC8MA,6B;MD9MA,wB;QAEwD,OAGF,cAAe,YAAV,IAAK,KAAK,EAAY,CCKn9C,cDrNgC,KCqNtB,KAAL,GAAiB,KAAtB,CDiN8C,MAAZ,CAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,qBAAE,YAAV,IAAK,KAAK,EAAM,KAAM,KAAX,CAAf,C;O;KAFtD,C;4FAGA,yB;MAqOA,kBAS6D,sB;MAT7D,+B;MArOA,wB;QAEwD,OgBAA,eAAW,ChB409B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgB508B,MAAK,UhBAK,KgBAO,KAAZ,CAAHb,C;O;KhBFxD,C;wFAIA,yB;MDmMA,6B;MC7LA,4C;MANA,wB;QAEqD,OAMD,WAAW,IAAX,EDoMjB,cC1M2B,KD0MjB,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KARpD,C;wFAGA,yB;MCiMA,6B;MD9LA,4C;MAHA,wB;QAEsD,OAGF,WAAW,IAAX,ECqMjB,cDxM4B,KCwMIB,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KALpD,C;wFAGA,yB;MAAA,4C;MAAA,wB;QAEoD,kBAAW,IAAX,EAaiB,KAAjB,C;O;KAFpD,C;wFAGA,yB;MAwNA,kBAS6D,sB;MAT7D,+B;MgBxNA,8C;MhBAA,wB;QAEsD,OgBAA,YhB+NjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgB/NiB,EhBAmb,KgBAnB,C;O;KhBFtD,C;wFAIA,yB;MDsLA,6B;MCxKA,kD;MAdA,wB;QAMqD,OAcD,cAAc,IAAd,ED2KjB,cCzL2B,KDyLjB,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KApBpD,C;wFAOA,yB;MCgLA,6B;MDzKA,kD;MAPA,wB;QAMsD,OAOFC,cAAc,IAAd,EC4KjB,cDnL4B,KCmLiB,KAAL,GAAiB,KAAtB,CD5KiB,C;O;KAbpD,C;wFAOA,yB;MAAA,kD;MAAA,wB;QAMoD,qBAAc,IAAd,EAaoB,KAApB,C;O;KANpD,C;wFAOA,yB;MA+LA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBAA,wB;QAMsD,OgBAA,ehBkMjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBIMiB,EhBAmb,KgBAnB,C;O;KhBNtD,C;kgAQA,yB;MDyJA,6B;MC7LA,4C;MAoCA,wB;QAMiD,OAxCG,WAAW,IAAX,EDoMjB,cC5J4B,KD4JIB,KAAL,GAAiB,GAAtB,CpMiB,C;O;KAKpD,C;kgAOA,yB;MCmJA,6B;MD9LA,4C;MA2CA,wB;QAMkD,OA/CE,WAAW,IAAX,ECqMjB,cDtJ6B,KCsJnB,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KAYCpD,C;kgAOA,yB;MAIDA,4C;MAkDA,wB;QAMgD,OAtDI,WAAW,IAAX,EAsDA,KAtDA,C;O;KAGpD,C;kgAOA,yB;MAkKA,kBAS6D,sB;MAT7D,+B;MgBxNA,8C;MhBsDA,wB;QAMkD,OgB1DI,YhB+NjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgB/NiB,EhB0DoB,KgB1DpB,C;O;KhBoDtD,C;wFAQA,yB;MD4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MC3LI,wB;QAQ6C,OD8LR,eAAW,OC5OI,cAAc,IAAd,ED2KjB,cC7HmB,KD6HT,KAAL,GAAiB,GAAtB,CC3KiB,CAkLf,KD0DW,CAX,C;O;KCtMrC,C;wFASA,yB;MCoha,6B;MDzKA,kD;MCwOJ,4B;MAAA,iC;MDnLI,wB;QAQ+C,OCsLR,gBAAY,QD7OC,cAAc,IAAd,EC4KjB,cDrHqB,KCqHX,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KD9LvC,C;wFASA,yB;MA9DA,kD;MA8DA,wB;QAQ2C,OAhES,cAAc,IAAd,EAgEL,KAhEK,C;O;KAwDpD,C;wFASA,yB;MA+HA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBgEA,wB;QAQ6C,OgBIES,ehBkMjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBIMiB,EhBkEU,KgBIEV,C;O;KhB0DtD,C;wEAUA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QAAY,C;O;KANzC,C;wEAQA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QAAY,C;O;KANzC,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAE6D,0BAAU,IAAV,EAAGB,KAAHb,C;O;KAF7D,C;wFAIA,yB;MAAA,6B;MAAA,2B;QAOmD,qBAAK,aAAS,QAAd,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,2B;QAOmD,qBAAK,cAAU,QAaf,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAC,KAAM,KAAZB,C;O;KAFjD,C;SFAGA,yB;MAAA,6B;MAAA,wB;QAEgD,qBAAK,IAAK,KAAL,GAAC,KAAM,KAAxB,C;O;KAFhD,C;wFAGA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAC,KA

AM,KAAzB,C;O;KAFjD,C;wEAGA,yB;MAAA,6B;MAAA,mB;QAEgC,qBAAU,CAAL,SAAL,C;O;KAFhC,C;8  
EAlA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,C;O;KAVxC,C;gFAWA,yB;MAAA,4B;MAAA,  
mB;QAUqC,OAAK,QAAL,SAAK,C;O;KAV1C,C;4EAWA,Y;MASiC,gB;K;8EACjC,yB;MAAA,kBASqD,sB;MA  
TrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,iB;O;KATnC,C;gFAWA,yB;MDwDJ,0B;MAAA,+B;MCxDI,mB;  
QASqC,OD0DA,eAAW,OC1DX,SD0DW,CAAX,C;O;KCnErC,C;kFAUA,yB;MC+CJ,4B;MAAA,iC;MD/CI,mB;  
QASuC,OCiDA,gBAAy,QDjDZ,SCiDY,CAAZ,C;O;KD1DvC,C;8EAUA,Y;MAEmC,W;K;gFACnC,yB;MAAA,k  
BAS6D,sB;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,iBAAN,C;O;KATrC,C;gFAWA,yB;  
MASA,gD;MATA,mB;QAQqC,OAoE,aAAa,SAAb,C;O;KAFvC,C;kFASA,yB;MAAA,gD;MAAA,mB;QAMuC,o  
BAAa,SAAb,C;O;KANvC,C;8BAQA,Y;MAAyC,OArDD,oBAAL,SAAK,CAAL,iBAqDe,W;K;,,,;8BAhWtD,Y;M  
AAA,c;MAG2D,qD;MAH3D,a;K;4BAAA,iB;MAAA,2IAG2D,oCAH3D,G;K;sEAOwA,yB;MAAA,6B;MAAA,4B  
;QAWwC,qBAAU,SAAV,C;O;KAXxC,C;wEAYA,yB;MAAA,6B;MAAA,4B;QAWyC,qBAAU,SAAV,C;O;KAX  
zC,C;wEAYA,yB;MAAA,6B;MAAA,4B;QAUuC,qBAAK,SAAL,C;O;KAVvC,C;wEAWA,yB;MAAA,6B;MAAA  
,4B;QAWwC,qBAAK,SAAK,QAAY,C;O;KAXxC,C;uEAaA,yB;MAAA,gD;MAAA,4B;QASyC,oBAAkB,SAAlB,  
C;O;KATzC,C;wEAUA,yB;MAAA,gD;MAAA,4B;QAS0C,oBAAa,SAAb,C;O;KAT1C,C;Igc3ZA,4B;MACqB,sB  
;K;sCAKjB,iB;MAM4C,OhCuXT,SgCvXS,aAAQ,KAAR,ChCuXT,C;K;sCgCrXnC,wB;MAOI,aAAQ,KAAR,IAA  
iB,KhCyQY,K;K;iFgCrQH,Y;MAAQ,OAAA,YAAQ,O;K;mCAE9C,Y;MAC6E,8BAAS,YAAT,C;K;IAGvD,mC;  
MAAgC,uB;MAA/B,oB;MACnB,eAAoB,C;K;2CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;4CACvC,Y;MAAwD,Q;  
MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OhCiWK,SgCjWmB,aAAM,mBAAN,EAAM,2BAAN,OhCiWnB,  
C;;QgCjWgD,MAAM,2BAAuB,YAAM,WAA7B,C;K;;yCAGzF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,OAAl,  
C;QAAgC,OAAO,K;MAEvC,OAAe,WAAR,YAAQ,EAAS,OhCmPK,KgCnPd,C;K;8CAGnB,oB;MACY,Q;MAA  
2B,gBAA3B,gE;MAA2B,c;;Qf0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,  
6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;Ue1nD6B,2Bf0nDR,Oe1nDQ,O;UAAA,W;YAAsB,oBAAR,YAAQ,  
Ef0nD9B,OjB34CJ,KgC/OkC,C;;Uf0nD7C,IAAI,OAAl,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;Me3nDH  
,iB;K;kCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,qC;MAAA,mD;MACgC,sBAAK,eA  
AS,IAAT,CAAL,C;MADhC,Y;K;;,mCAPI,Y;MAAA,OAKqB,oDALrB,M;K;mCAAA,Y;MAAA,c;MAKqB,wD;  
MALrB,a;K;iCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;8EAyDA,yB;MAAA,uC;MAWoC,wC;QAAA,wB;UAAW,  
OAAA,aAAK,KAAL,ChC8NV,K;S;O;MgCzOrC,6B;QAWI,OAAO,mBAAU,gCAAS,IAAT,GAAe,sBAAf,CAAV  
,C;O;KAXX,C;gFAcA,oB;MAGkE,e;K;I6LnE5C,wC;MASBiB,iC;MAtBsD,2BAAgB,KAAhB,EAAuB,YAAvB,E  
AAqC,CAArC,C;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;2CAExC,iB;MAA8C,W7NwCoB,Y6NxCP  
B,U7NwCqC,KAAjB,E6NxCX,K7NwCwC,KAA7B,C6NxCPB,K;MAAA,S;QAAkB,O7NwCE,Y6NxCF,K7NwC  
mB,KAAjB,E6NxC0,S7NwCsB,KAA7B,C6NxCF,K;;MAAIB,W;K;kCAE9C,Y;MAKkC,O7NiCgC,Y6NjChC,U7  
NiCiD,KAAjB,E6NjCxB,S7NiCqD,KAA7B,C6NjChC,I;K;iCAEIC,iB;MAEY,UAAwB,M;MADhC,2CAAuB,kBA  
Aa,KAAM,UAAAnB,KACf,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAAd,QAAxB,CADe,CAAvB,C;K;m  
CAGJ,Y;MACI,OAAl,cAAJ,GAAe,EAAf,GAAwB,MAAK,U7NyQA,K6NzQL,QAAqB,S7NyQhB,K6NzQL,I;K;  
mCAE5B,Y;MAAkC,OAAE,UAAf,qBAAU,S;K;IAE5C,+B;MAAA,mC;MACI,aAC8B,cAAU,4BAAK,UAAf,EA  
A0B,4BAAK,UAA/B,C;K;;IAFIC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;IAYJ,oD;MA4CI,uC;MAtCI,IAAI,S  
AAQ,CAAZ,C;QAAuB,MAAa,gCAAYB,wBAAzB,C;MACpC,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAYB,w  
EAAzB,C;MAG5C,aAGyB,K;MAEZB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YA  
GuB,I;K;yCAEvB,Y;MAAgD,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAEHd,Y;MAMqC,OA  
AI,YAAO,CAAX,G7NvB6B,Y6NuBf,U7NvBgC,KAAjB,E6NuBP,S7NvBoC,KAA7B,C6NuBf,IAAd,G7NvB6B,  
Y6NuBG,U7NvBc,KAAjB,E6NuBW,S7NvBkB,KAA7B,C6NuBG,I;K;uCAErE,iB;MAEY,UAAwB,M;MADhC,i  
DAA6B,kBAAa,KAAM,UAAAnB,KACrB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAAd,QAAxB,KAA8  
C,cAAQ,KAAM,KADvC,CAA7B,C;K;yCAGJ,Y;MACI,OAAl,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,U7Ni  
NN,K6NjNC,QAAqB,S7NiNtB,K6NjNC,IAAN,SAAGD,SAAhD,I;K;yCAE5B,Y;MAAkC,OAAl,YAAO,CAAX,G  
AAgB,UAAf,qBAAU,SAAV,cAAqB,SAAnC,GAAgD,UAAf,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;  
K;IAEHf,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,UAAhB,EAA4B,QAA5B,EAA5C,IAAtC,C;K;;IAT/F,iD;  
MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAoBiC,oD;MAAuC,uB;MACxE,sBAA2B,I;MAC3B,iBAAmC,OAAO,CA  
A1C,G7NxDe,Y6NwDrB,K7NxDsC,KAAjB,E6NwDZ,I7NxDyC,KAA7B,C6NwDrB,KAA7C,G7NxDe,Y6Nw

DF,K7NxDMB,KAAjB,E6NwDO,I7NxDSB,KAA7B,C6NwDF,K;MACHe,c7N2RmC,S6N3RhB,I7N2RgB,C;M6N  
1RnC,cAAuB,cAAJ,GAAa,KAAb,GAawB,mB;K;gDAE3C,Y;MAAKC,qB;K;iDAEIC,Y;MACI,YAAY,W;MACZ,  
IAAI,6BAAS,mBAAT,QA AJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAEV,c7NID6  
C,S6Nkd7C,W7NIDuD,KAAK,G6NkdP,D,W7NID+D,KAAx,IAAf,C;;M6NoDjD,OAAO,K;K;;IC3Hf,yB;K;mCA  
II,Y;MAA4B,uB;K;;IAMhC,0B;K;oCAII,Y;MAA4B,wB;K;;IAMhC,wB;K;kCAII,Y;MAA4B,sB;K;;IAMhC,yB;K;  
mCAII,Y;MAA4B,uB;K;;I9M5BP,qB;MAErB,6B;MAFwD,gB;K;IAExD,2B;MAAA,+B;MACI,iBAGoC,a;MAEp  
C,iBAGoC,c;MAEpC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANbtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;sGAs  
BA,yB;MjBqRA,WAS6D,wB;MAT7D,+B;MiB7PA,gD;MAxBA,wB;QAM0D,OAYBS,aAAa,IAAK,KAAIB,EAA  
8B,CjB+P5D,eAAW,oBiBxRyB,KjBwR9B,KAAK,CAAL,UAAN,CiB/P4D,MAA9B,C;O;KA/BnE,C;sGAQA,yB;  
Mf8QA,aAS6D,0B;MAT7D,+B;Me9PA,gD;MAhBA,wB;QAM2D,OaiBQ,aAAa,IAAK,KAAIB,EAA8B,CfgQ5D,  
eAAW,oBeJR0B,KfiR/B,KAAK,CAAL,YAAN,CehQ4D,MAA9B,C;O;KA/vBnE,C;sGAQA,yB;MhByRA,kBAS6D  
,sB;MAT7D,+B;MgBjRA,gD;MARA,wB;QAMyD,OASU,aAAa,IAAK,KAAIB,EAA8B,ChBmR5D,eAAW,oBgB5  
RwB,KhB4R7B,KAAK,CAAL,iBAAN,CgBnR4D,MAA9B,C;O;KafnE,C;kGAQA,yB;MAAA,gD;MAAA,wB;Q  
AOmE,oBAAa,IAAK,KAAIB,EAAwB,KAAM,KAA9B,C;O;KAPnE,C;4FASA,yB;MjBoPA,WAS6D,wB;MAT7D  
,+B;MiBpPA,wB;QAEuD,OASA,eAAM,IAAK,KAAK,KAAW,CjBkP7C,eAAW,oBiB3PiB,KjB2PtB,KAAK,CAA  
L,UAAN,CiBIP6C,MAAX,CAAhB,C;O;KAXvD,C;4FAGA,yB;MfkPA,aAS6D,0B;MAT7D,+B;MelPA,wB;QAE  
wD,OAMD,eAAM,IAAK,KAAK,KAAW,CfmP7C,eAAW,oBezPkB,KfyPvB,KAAK,CAAL,YAAN,CenP6C,MA  
AX,CAAhB,C;O;KARvD,C;4FAGA,yB;MhBkQA,kBAS6D,sB;MAT7D,+B;MgBIQA,wB;QAEsD,OAGC,eAAM,  
IAAK,KAAK,KAAW,ChBsQ7C,eAAW,oBgBzQgB,KhByQrB,KAAK,CAAL,iBAAN,CgBtQ6C,MAAX,CAAhB,  
C;O;KALvD,C;4FAGA,yB;MAAA,+B;MAAA,wB;QAEuD,sBAAM,IAAK,KAAK,KAAK,KAAM,KAAX,CAAh  
B,C;O;KAFvD,C;8FAIA,yB;MjBuOA,WAS6D,wB;MAT7D,+B;MiBvOA,wB;QAEwD,OASA,eAAM,IAAK,KAA  
K,UAAY,CjBqO/C,eAAW,oBiB9OmB,KjB8OxB,KAAK,CAAL,UAAN,CiBrO+C,MAAZ,CAAhB,C;O;KAXxD,  
C;8FAGA,yB;MfqOA,aAS6D,0B;MAT7D,+B;MerOA,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAY,CfsO/C,e  
AAW,oBe5OoB,Kf4OzB,KAAK,CAAL,YAAN,CetO+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MhBqPA,kB  
AS6D,sB;MAT7D,+B;MgBrPA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,UAAY,ChByP/C,eAAW,oBgB5PkB,Kh  
B4PvB,KAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QA  
EwD,sBAAM,IAAK,KAAK,UAAM,KAAM,KAAZ,CAAhB,C;O;KAFxD,C;8FAIA,yB;MjB0NA,WAS6D,wB;M  
AT7D,+B;MiB1NA,wB;QAEwD,OASA,eAAM,IAAK,KAAK,UAAY,CjBwN/C,eAAW,oBiBjOmB,KjBiOxB,KA  
AK,CAAL,UAAN,CiBxN+C,MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;MfwNA,aAS6D,0B;MAT7D,+B;MexN  
A,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAY,CfyN/C,eAAW,oBe/NoB,Kf+NzB,KAAK,CAAL,YAAN,Cez  
N+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MhBwOA,kBAS6D,sB;MAT7D,+B;MgBxOA,wB;QAEuD,OA  
GC,eAAM,IAAK,KAAK,UAAY,ChB4O/C,eAAW,oBgB/OkB,KhB+OvB,KAAK,CAAL,iBAAN,CgB5O+C,MAA  
Z,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,sBAAM,IAAK,KAAK,UAAM,KAAM,K  
AAZ,CAAhB,C;O;KAFxD,C;0FAIA,yB;MjB6MA,WAS6D,wB;MAT7D,+B;MiBpMA,8C;MATA,wB;QAEsD,O  
ASA,YAAY,IAAZ,EjB2MjB,eAAW,oBiBpNe,KjBoNpB,KAAK,CAAL,UAAN,CiB3MiB,C;O;KAXtD,C;0FAGA  
,yB;Mf2MA,aAS6D,0B;MAT7D,+B;MerMA,8C;MANA,wB;QAEuD,OAMD,YAAY,IAAZ,Ef4MjB,eAAW,oBel  
NgB,KfkNrB,KAAK,CAAL,YAAN,Ce5MiB,C;O;KARtD,C;0FAGA,yB;MhB2NA,kBAS6D,sB;MAT7D,+B;MgB  
xNA,8C;MAHA,wB;QAEqD,OAGC,YAAY,IAAZ,EhB+NjB,eAAW,oBgBIOc,KhBkOnB,KAAK,CAAL,iBAAN,  
CgB/NiB,C;O;KALtD,C;0FAGA,yB;MAAA,8C;MAAA,wB;QAEsD,mBAAY,IAAZ,EAAkB,KAAIB,C;O;KAFtD  
,C;0FAIA,yB;MjBgMA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MArBA,wB;QAMsD,OAqBA,eAAe,IAAf,EjB8Kj  
B,eAAW,oBiBnMe,KjBmMpB,KAAK,CAAL,UAAN,CiB9KiB,C;O;KA3BtD,C;0FAOA,yB;Mf0LA,aAS6D,0B;M  
AT7D,+B;Me5KA,oD;MAdA,wB;QAMuD,OAcD,eAAe,IAAf,Ef+KjB,eAAW,oBe7LgB,Kf6LrB,KAAK,CAAL,Y  
AAN,Ce/KiB,C;O;KApBtD,C;0FAOA,yB;MhBsMA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MAPA,wB;QAMqD,  
OAOc,eAAe,IAAf,EhBkMjB,eAAW,oBgBzMc,KhByMnB,KAAK,CAAL,iBAAN,CgBIMiB,C;O;KAbtD,C;0FAO  
A,yB;MAAA,oD;MAAA,wB;QAMsD,sBA Ae,IAAf,EAAqB,KAArB,C;O;KANtD,C;oGAQA,yB;MjBmKA,WAS6  
D,wB;MAT7D,+B;MiBpMA,8C;MAiCA,wB;QAMkD,OArCI,YAAY,IAAZ,EjB2MjB,eAAW,oBiBtKgB,KjBsKr  
B,KAAK,CAAL,UAAN,CiB3MiB,C;O;KA+BtD,C;oGAOA,yB;Mf6JA,aAS6D,0B;MAT7D,+B;MerMA,8C;MAw  
CA,wB;QAMmD,OA5CG,YAAY,IAAZ,Ef4MjB,eAAW,oBehKiB,KfgKtB,KAAK,CAAL,YAAN,Ce5MiB,C;O;K

AsCtD,C;oGAOA,yB;MhByKA,kBAS6D,sB;MAT7D,+B;MgBxNA,8C;MA+CA,wB;QAMiD,OAnDK,YAAy,IAAZ,EhB+NjB,eAAW,oBgB5Ke,KhB4KpB,KAAK,CAAL,iBAAN,CgB/NiB,C;O;KA6CtD,C;oGAOA,yB;MatDA,8C;MAsDA,wB;QAMkD,OA1DI,YAAy,IAAZ,EA0DA,KA1DA,C;O;KAoDtD,C;0FAQA,yB;MjBsIA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MjB4OJ,0B;MAAA,+B;MiBvMI,wB;QAQ6C,OjB0MP,eAAW,OiBjPK,eAAe,IAAf,EjB8KjB,eAAW,oBiBvIM,KjBuIX,KAAK,CAAL,UAAN,CiB9KiB,CA4KjB,KjBqEY,SAAX,C;O;KiBINtC,C;0FASa,yB;Mf8HA,aAS6D,0B;MAT7D,+B;Me5KA,oD;Mf6OJ,4B;MAAA,iC;Me/LI,wB;QAQ+C,OfkMP,gBAAY,QelPE,eAAe,IAAf,Ef+KjB,eAAW,oBe/HQ,Kf+Hb,KAAK,CAAL,YAAN,Ce/KiB,CAsLf,Kf4Da,SAAZ,C;O;Ke1MxC,C;0FASA,yB;MhBwIA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBkQJ,6B;MgB3MI,wB;QAQ2C,OhB8MP,cgBvQkB,eAAe,IAAf,EhBkMjB,eAAW,oBgBzII,KhByIT,KAAK,CAAL,iBAAN,CgBIMiB,CAGMnB,KhBuEW,QAAV,C;O;KgBtNpC,C;0FASA,yB;MAhEA,oD;MAGeA,wB;QAQ6C,OAIES,eAAe,IAAf,EAkEL,KAIEK,C;O;KA0DtD,C;0EAUA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;gQAQA,yB;MAAA,gD;MAAA,wB;QAE+D,2BAAW,IAAX,EAAiB,KAAjB,C;O;KAF/D,C;0FAIA,yB;MAAA,+B;MAAA,2B;QAOoD,sBAAM,oBAAS,QAAT,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,2B;QAOoD,sBAAM,6BAAU,QAAV,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAC,KAAM,KAAPB,CAAN,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;MAAA,wB;QAEkD,sBAAM,IAAK,KAAL,IAAA,KAAM,KAANB,CAAN,C;O;KAFID,C;0FAGA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAC,KAAM,KAAPB,CAAN,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;MAAA,mB;QAEiC,sBAAM,SAAK,MAAX,C;O;KAFjC,C;gFAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,S;O;KAVxC,C;kFAWA,yB;MAAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK,S;O;KAV1C,C;8EAWA,Y;MAU1C,OAAA,SAAK,Q;K;gFACtC,Y;MASmC,gB;K;kFAEnC,yB;MjBmEJ,0B;MAAA,+B;MiBnEI,mB;QASqC,OjBqEC,eAAW,OiBrEZ,SjBqEY,SAAX,C;O;KiB9EtC,C;oFAUA,yB;Mf0DJ,4B;MAAA,iC;Me1DI,mB;QASuC,Of4DC,gBAAY,Qe5Db,Sf4Da,SAAZ,C;O;KerExC,C;gFAUA,yB;MhBqEJ,6B;MgBrEI,mB;QASmC,OhBuEC,cgBvED,ShBuEW,QAAV,C;O;KgBhFpC,C;kFAUA,Y;MAEQC,W;K;kFAErC,yB;MASA,kD;MATA,mB;QAQqC,OASE,cAAc,SAAd,C;O;KAjBvC,C;oFASA,yB;MAAA,kD;MAAA,mB;QAQuC,qBAAC,SAAd,C;O;KARvC,C;+BAUA,Y;MAAyC,qBAAC,SAAd,C;K;+;+BAnW7C,Y;MAAA,c;MAG4D,qD;MAH5D,a;K;6BAAA,iB;MAAA,2IAG4D,oCAH5D,G;K;wEAuWA,yB;MAAA,+B;MAAA,4B;QAW0C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX1C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAW2C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAWyC,sBAAW,oBAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;yEAYA,yB;MAAA,kD;MAAA,4B;QAS2C,qBAAmB,SAANB,C;O;KAT3C,C;0EAUA,yB;MAAA,kD;MAAA,4B;QAS4C,qBAAC,SAAd,C;O;KAT5C,C;liB9ZA,6B;MACqB,sB;K;uCakjB,iB;MAM6C,OjBsYP,UiBtYO,aAAQ,KAAR,CjBsYP,C;K;uCIBpYtC,wB;MAOI,aAAQ,KAAAR,IAAiB,KjBoRc,K;K;kFiBhRL,Y;MAAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAGxD,oC;MAAiC,wB;MAAhC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;8CACvC,Y;MAAyD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjBgXO,UiBhXiB,aAAM,mBAAN,EAAM,2BAAN,OjBgXjB,C;;QiBhX+C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;0CAG3F,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QA AJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OjB8PO,KiB9PhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QhB0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAp,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UgB1nD6B,2BhB0nDR,OgB1nDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,EhB0nD/B,ODh4CF,KiB1PiC,C;;UhB0nD9C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MgB3nDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,sC;MAAA,oD;MACgC,uBAAK,iBAAU,IAAV,CAAL,C;MADhC,Y;K;;;oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAyDA,yB;MAAA,yC;MAWSc,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjByOV,K;S;O;MiBpPvC,6B;QAWI,OAAO,oBAAW,kBAAU,IAAV,EAAGB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;I8LnE9C,2C;MAsBnB,kC;MatByD,4BAAiB,KAAjB,EAAwB,YAAxB,K;K;qFAC/B,Y;MAAQ,iB;K;4FACD,Y;MAAQ,gB;K;8CAEzC,iB;MAA+C,W/MgDoB,a+MhDpB,U/MgDsC,KAAIB,E+MhDX,K/MgDyC,KAA9B,C+MhDpB,K;MAAA,S;QAAkB,O/MgDE,a+MhDF,K/MgDoB,KAAIB,E+MhDO,S/MgDuB,KAA9B,C+MhDF,K;;MAAIB,W;K;qCAE/C,Y;MAKkC,O/MyCiC,a+MzCjC,U/MycmD,KAAIB,E+MzCzB,S/MyCuD,KAA9B,C+MzCjC,I;K;oCAEIC,iB;MAEY,UAAwB,M;MADhC,8CAAwB,k

BAAa,KAAM,UAAAnB,KACbB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,CADgB,CAAxB  
,C;K;sCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,M/M0QK,CArCkB,U+MrOjB,U/MqO4B,KAAL,KAAo  
B,CAVzB,U+M3NP,U/M2Na,yB+M3NH,E/M2NG,CAAN,CAUyB,MAApB,CAAN,CAqCIB,MAAK,Q+M1QV,Q  
/M0QK,CArCkB,U+MrOoB,S/MqOT,KAAL,KAAoB,CAVzB,U+M3N6B,S/M2NvB,yB+M3NgC,E/M2NhC,CAA  
N,CAUyB,MAApB,CAAN,CAqCIB,MAAK,Q+M1QV,I;K;sCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5  
C,gC;MAAA,oC;MACI,aAC+B,iBAAW,6BAAM,UAAjB,EAA4B,6BAAM,UAAIC,C;K;;;IAFnC,4C;MAAA,2C;  
QAAA,0B;;MAAA,oC;K;;IAYJ,qD;MA4CI,wC;MAtCI,IAAI,gBAAJ,C;QAAwB,MAAA,gCAAyB,wBAAzB,C;M  
ACrC,IAAI,sCAAJ,C;QAA4B,MAAA,gCAAyB,yEAAzB,C;MAG7C,aAG0B,K;MAE1B,YAGyB,4BAA0B,KAA1  
B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEzB,YAGwB,I;K;0CAExB,Y;MAAiD,oCAAyB,UAAzB,EAAgC,SAAhC  
,EAAcC,SAAtC,C;K;yCAEjD,Y;MAMqC,OAAI,uBAAO,CAAX,G/Mf8B,a+MehB,U/MfkC,KAAIB,E+MeR,S/Mf  
sC,KAA9B,C+MehB,IAAd,G/Mf8B,a+MeE,U/MfgB,KAAIB,E+MeU,S/MfoB,KAA9B,C+MeE,I;K;wCAErE,iB;  
MAEY,UAAwB,M;MADhC,kDAA8B,kBAAa,KAAM,UAAAnB,KACtB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,  
KAAM,KAAd,QAAxB,KAA8C,kBAAQ,KAAM,KAAd,CADxB,CAA9B,C;K;0CAGJ,Y;MACI,OAAI,cAAJ,GAA  
e,EAAf,GAAwB,OAAM,M/MkND,CArCkB,U+M7KX,U/M6KsB,KAAL,KAAoB,CAVzB,U+MnKD,U/MmKO,y  
B+MnKG,E/MmKH,CAAN,CAUyB,MAApB,CAAN,CAqCIB,MAAK,Q+MINJ,Q/MkND,CArCkB,U+M7K0B,S/  
M6Kf,KAAL,KAAoB,CAVzB,U+MnKmC,S/MmK7B,yB+MnKsC,E/MmKtC,CAAN,CAUyB,MAApB,CAAN,C  
AqCIB,MAAK,Q+MINJ,IAAN,SAAqF,cAAU,6BAAU,EAAV,CAAV,CAAyB,QAA9G,I;K;0CAE5B,Y;MAAkC,  
OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAArB,WAAAd,GAAgD,UAAF,2BAAgB,SAAhB,c  
AA6B,SAAD,aAA5B,W;K;IAEhF,sC;MAAA,0C;K;mEACI,sC;MAQ+F,4BAAiB,UAAjB,EAA6B,QAA7B,EAAu  
C,IAAvC,C;K;;;IATnG,kD;MAAA,iD;QAAA,gC;;MAAA,0C;K;;;IAoBkC,qD;MAA0C,wB;MAC5E,sBAA2B,I;M  
AC3B,iBAAMC,kBAAO,CAA1C,G/MhDmE,a+MgDtB,K/MhDwC,KAAIB,E+MgDb,I/MhD2C,KAA9B,C+MgDt  
B,KAA7C,G/MhDmE,a+MgDH,K/MhDqB,KAAIB,E+MgDM,I/MhDwB,KAA9B,C+MgDH,K;MACHe,c/M0SsC,  
U+M1SnB,I/M0SmB,C;M+MzStC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;iDAE3C,Y;MAAkC,qB;K;mDAE  
C,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAA,6B;QAC  
3B,iBAAU,K;;QAEV,c/M/C+C,U+M+C/C,W/M/C0D,KAAK,K+M+CvD,W/M/CkE,KAAX,CAAhB,C;;M+MiDn  
D,OAAO,K;K;;wEC7Hf,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wE  
AUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8  
C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAO  
I,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;oFC7BA,yB;MAAA,gD;MAAA,4B;QAM6C,OAAQ,ajO+R  
hB,ciO/RgB,C;O;KANrD,C;oGAQA,yB;M/GwCA,iB;M+GxCA,4B;QAMqD,O/GwCM,MAAO,OIH+O7B,ckH/O  
6B,C;O;K+G9CIE,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAQ,sBjO+QzB,ciO/QyB,C;O;KAN9D,C;8FA  
QA,yB;MAAA,0D;MjOwWA,6B;MiOxWA,4B;QAOmD,OjO2WZ,ciO3WoB,kBjOsQtB,ciOtQsB,CjO2WpB,C;O;  
KiOIXvC,C;4FASA,yB;MAAA,wD;MjO+VA,6B;MiO/VA,4B;QAOkD,OjOkWX,ciOIWmB,iBjO6PrB,ciO7PqB,C  
jOkWnB,C;O;KiOzWvC,C;gFASA,yB;MAAA,4C;MjOsVA,6B;MiOtVA,sC;QAayD,OjOmVIB,ciOnV0B,WjO8O  
5B,ciO9O4B,EAAW,QAAX,CjOmV1B,C;O;KiOhWvC,C;kFAgBA,yB;MAAA,8C;MjOsUA,6B;MiOtUA,sC;QAa  
0D,OjOmUnB,ciOnU2B,YjO8N7B,ciO9N6B,EAAy,QAAZ,CjOmU3B,C;O;KiOhVvC,C;oFAgBA,yB;MAAA,gD;  
MAAA,4B;QAM8C,OAAS,ajNgOhB,ciNhOgB,C;O;KANvD,C;oGAQA,yB;MAAA,gE;MAAA,4B;QAMsD,OAA  
S,qBjNwNxB,ciNxNwB,C;O;KAN/D,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMuD,OAAS,sBjNgNzB,ciNhNyB,  
C;O;KANhE,C;8FAQA,yB;MAAA,0D;MjN6SA,+B;MiN7SA,4B;QAOqD,OjNgTX,eiNhToB,kBjNuMvB,ciNvMu  
B,CjNgTpB,C;O;KiNvT1C,C;4FASA,yB;MAAA,wD;MjNoSA,+B;MiNpSA,4B;QAOoD,OjNuSV,eiNvSmB,iBjN  
8LtB,ciN9LsB,CjNuSnB,C;O;KiN9S1C,C;+EASA,yB;MAAA,4C;MjN2RA,+B;MiN3RA,sC;QAa2D,OjNwRjB,ei  
NxR0B,WjN+K7B,ciN/K6B,EAAW,QAAX,CjNwR1B,C;O;KiNrS1C,C;iFAeA,yB;M/GgEA,4C;MIG4MA,+B;Mi  
N5QA,sC;QAa4D,OjNyQIB,ekGzMuB,WIGgG1B,ckGhG0B,EAAW,C+GhEK,Q/GgEL,IAAX,CIGyMvB,C;O;Ki  
Nr1C,C;oFAeA,yB;MIOWJI,6B;MkO1SJ,gD;MAKJA,4B;QAM8C,OAIJO,ajO+RhB,CDcE,cAAU,cAAL,GAAiB,  
GAAtB,CCdF,MIo/RgB,C;O;KA4IrD,C;oGAQA,yB;M/G1GA,iB;M+G0GA,4B;QAMsD,O/G1GK,MAAO,OnHu  
M3B,c2N1Ge,GAAY,GxG7FA,CwG6Fb,GAA6C,EAA7C,I;O;KOOzE,C;8FAQA,yB;MAAA,0D;MIO+LA,0B;MAAA,+B;M  
kO/LA,4B;QAOqD,OIOmMZ,eAAW,OkOnMS,kBIOgGnB,cAAL,GAAiB,GkOhGO,CIOmMT,CAAX,C;O;KkO1

MzC,C;4FASA,yB;MAAA,wD;MIOsLA,0B;MAAA,+B;MkOtLA,4B;QAOoD,OIO0LX,eAAW,OkO1LQ,iBlOuFl  
B,cAAL,GAAiB,GkOvFM,CIO0LR,CAAX,C;O;KkOjMzC,C;gFAUA,yB;MAAA,4C;MIOqJA,+B;MkOrJA,sC;Q  
Aa2D,OIOkJb,ekOIJ0B,WIOmD7B,ckOnD6B,EAAW,QAAX,CIOk1B,C;O;KkO/J1C,C;kFAeA,yB;MAAA,8C;  
MIOsIA,+B;MkOtIA,sC;QAa4D,OIOmIIB,ekOnI2B,YIOoC9B,ckOpC8B,EAAAY,QAAX,CIOmI3B,C;O;KkOhJ1C,  
C;oFAeA,yB;MhOgFI,6B;MgO3SJ,gD;MA2NA,4B;QAM+C,OA3NM,ajO+RhB,CCeE,cAAU,cAAL,GAAiB,KA  
AtB,CDfF,MiO/RgB,C;O;KAqNrD,C;oGAQA,yB;M/GnLA,iB;M+GmLA,4B;QAMuD,O/GnLI,MAAO,OjHkNzB,  
cyN3CpC,GAAY,KxGvKiD,CwGuK9D,GAA+C,EAA/C,I;O;KOMJ,C;sGAQA,yB;MPZA,kE;MOYA,4B;QAMw  
D,OPZoB,sBzNmCnC,cyNnCe,GAAW,KAAS,C;O;KOM5E,C;8FAQA,yB;MAAA,0D;MhOuHA,4B;MAAA,iC;M  
gOvHA,4B;QAOUd,OhO2HZ,gBAAy,QgO3HQ,kBhOwBrB,cAAL,GAAiB,KgOxBS,ChO2HR,CAAZ,C;O;KgOI  
I3C,C;4FASA,yB;MAAA,wD;MhO8GA,4B;MAAA,iC;MgO9GA,4B;QAOSD,OhOkHX,gBAAy,QgOIHO,iBhOep  
B,cAAL,GAAiB,KgOfQ,ChOkHP,CAAZ,C;O;KgOzH3C,C;gFAUA,yB;MAAA,4C;MhOyFA,iC;MgOzFA,sC;QA  
a6D,OhOsFhB,gBgOf0B,WhOX9B,cgOW8B,EAAW,QAAX,ChOsF1B,C;O;KgOnG7C,C;kFAeA,yB;MAAA,8C;  
MhO0EA,iC;MgO1EA,sC;QAa8D,OhOuEjB,gBgOvE2B,YhO1B/B,cgO0B+B,EAAy,QAAX,ChOuE3B,C;O;KgO  
pF7C,C;ICtRA,qC;MAEI,SIOuIoD,ckOvI3C,CIOuI2C,EkOvIvC,CIOuIuC,C;MkOtIpD,SIOsIoD,ckOtI3C,CIOsI2C,  
EkOtlvC,CIOsIuC,C;MkOrIpD,OIOmDkE,YkOnDvD,EIOmDwE,KAAjB,EkOnDjD,EIOmD8E,KAA7B,CkOnDv  
D,KAAAX,GIOkFsD,SkOIFjC,EIOkF2C,KAAK,GkOIF3C,EIOkFuD,KAAZ,IAAf,CkOIFtD,GIOqEqD,SAAU,CAAT  
,SkOIFpB,EIOkF8B,KAAK,GkOIF9B,EIOkF0C,KAAZ,IAAf,CABs,MAAK,GkOrExB,CIOqEmC,KAAAX,IAAf,C;  
K;IkOIEzD,qC;MACI,SINwIsD,ekNxI7C,CINwI6C,EkNxIzC,CINwIyC,C;MkNvItD,SINuIsD,ekNvI7C,CINuI6C,E  
kNvIzC,CINuIyC,C;MkNtItD,OINqDmE,akNrDxD,EINqD0E,KAAIB,EkNrDID,EINqDgF,KAA9B,CkNrDxD,KA  
AX,GIN+EwD,UkN/EnC,EIN+E8C,KAAK,UkN/E9C,EIN+E0D,KAAZ,CAAhB,CkN/ExD,GINkEuD,UAAW,CAa  
V,UkN/EtB,EIN+EiC,KAAK,UkN/EjC,EIN+E6C,KAAZ,CAAhB,CABU,MAAK,KkNIE3B,CINkEsC,KAAAX,CAA  
hB,C;K;IkN/D3D,uD;MAMBI,WAAO,CAAP,C;QAD8E,OIOwBZ,YkOvBID,KIOuBmE,KAAjB,EkOvBzC,GIOuB  
sE,KAA7B,CkOvBID,KAD8D,GACHD,GADgD,GIOuDxB,SkOtDf,GIOsDyB,KAAK,GkOtDxB,mBAAiB,GAAjB  
,EAAAsB,KAAAtB,EIO2WV,SkO3WuC,IIO2WvC,CkO3WU,CIOsDoC,KAAZ,IAAf,C;akOrDtD,WAAO,CAAP,C;Q  
AF8E,OIOwBZ,YkOtBID,KIOsBmE,KAAjB,EkOtBzC,GIOsBsE,KAA7B,CkOtBID,KAF8D,GAEhD,GAFgD,GIO  
0CzB,SkOxCd,GIOwCwB,KAAK,GkOxCvB,mBAAiB,KAAjB,EAAwB,GAAxB,EIO0WV,SkO1WwC,CAAC,IA  
AD,IIO0WxC,CkO1WU,CIOwCkC,KAAAX,IAAf,C;;QkOvC7C,MAAA,gCAAYB,eAAzB,C;K;IAGzB,uD;MAMBI,  
sBAAO,CAAP,C;QADkF,OINQf,akNPnD,KINOqE,KAAIB,EkNP1C,GINowE,KAA9B,CkNPnD,KADkE,GACp  
D,GADoD,GINkC1B,UkNjCjB,GINiC4B,KAAK,UkNjC3B,mBAAiB,GAAjB,EAAAsB,KAAAtB,EINkWP,UkNIWo  
C,IINkWP,C,CkNIWO,CINiCuC,KAAZ,CAAhB,C;akNhCxD,sBAAO,CAAP,C;QAFkF,OINQf,akNNnD,KINMqE,  
KAAIB,EkNN1C,GINMwE,KAA9B,CkNNnD,KAFkE,GAEpD,GAFoD,GINqB3B,UkNnBhB,GINmB2B,KAAK,  
KkNnB1B,mBAAiB,KAAjB,EAAwB,GAAxB,EINiWP,UkNjWsC,IAAD,alNiWrC,CkNjWO,CINmBqC,KAAAX,C  
AAhB,C;;QkNIB/C,MAAA,gCAAYB,eAAzB,C;K;IjOIDC,sB;MAEtB,8B;MAFyD,gB;K;IAEzD,4B;MAAA,gC;M  
ACI,iBAGqC,WAAO,CAAP,C;MAErC,iBAGqC,WAAO,MAAP,C;MAErC,kBAGmC,C;MAEnC,iBAGkC,E;K;;I  
AnBtC,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;wGAsBA,iB;MAM0D,OAAa,0BA6OjC,SAAL,GAAiB,KA7OqB  
,EAAU,KF4O3C,KAAL,GAAiB,GE5OqB,C;K;oGAEvE,iB;MAOoE,OAAa,0BAoO3C,SAAL,GAAiB,KApO+B,E  
AAU,KAOOrD,KAAL,GAAiB,KApO+B,C;K;wGAejF,yB;MA2PA,6B;MD5PA,8C;MCCA,wB;QAMyD,ODAS,Y  
AAiB,CC8PhD,cAAU,SAAL,GAAiB,KAAAtB,CD9PgD,MAAJB,ECAe,KDAc,KAA7B,C;O;KCNIE,C;wGAQA,yB  
;MA6PA,aAS6D,0B;MAT7D,+B;Me9PA,gD;MfCA,wB;QAM0D,OeAS,aAAkB,CfgQhD,eAAW,oBAAL,SAAK,  
CAAL,YAAN,CehQgD,MAAIB,EfAgB,KeAc,KAA9B,C;O;KfNnE,C;8FAQA,yB;MA2OA,6B;MA3OA,wB;QAE  
sD,ODMD,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAAtB,CD5O4B,MAAK,GAAW,CD2O5C,cEjPsC,KFiP5B,K  
AAL,GAAiB,GAAAtB,CC3O4C,MAAX,IAAf,C;O;KCRrD,C;8FAGA,yB;MAwOA,6B;MAxOA,wB;QAEuD,ODG  
F,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAAtB,CD5O4B,MAAK,GAAW,CC4O5C,cA/OuC,KA+O7B,KAAL,G  
AAiB,KAAAtB,CD5O4C,MAAX,IAAf,C;O;KCLrD,C;8FAGA,yB;MAqOA,6B;MArOA,wB;QAEqD,ODAA,cAAU  
,CC4O5B,cAAU,SAAL,GAAiB,KAAAtB,CD5O4B,MAAK,GCAI,KDAO,KAAAX,IAAf,C;O;KCFrD,C;8FAGA,yB;  
MA4OA,aAS6D,0B;MAT7D,+B;MA5OA,wB;QAEuD,OeAA,eAAW,CfmP7B,eAAW,oBAAL,SAAK,CAAL,YA  
AN,CenP6B,MAAK,KfAI,KeAO,KAAAX,CAAhB,C;O;KfFvD,C;gGAIA,yB;MA8NA,6B;MA9NA,wB;QAEuD,O  
DMD,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAAtB,CD/N6B,MAAK,GAAy,CD8N9C,cEpOwC,KFoO9B,KAA

L,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KCRtD,C;gGAGA,yB;MA2NA,6B;MA3NA,wB;QAEwD,ODGF,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAtB,CD/N6B,MAAK,GAAy,CC+N9C,cAlOyC,KAKO/B,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KCLtD,C;gGAGA,yB;MAwNA,6B;MAxNA,wB;QAEsD,ODAA,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAtB,CD/N6B,MAAK,GCAK,KDAO,KAAZ,IAAf,C;O;KCFtD,C;gGAGA,yB;MA+NA,aAS6D,0B;MAT7D,+B;MA/NA,wB;QAEwD,OeAA,eAAW,CfsO9B,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce tO8B,MAAK,UfAK,KeAO,KAAZ,CAAhB,C;O;KfFxD,C;gGAIA,yB;MAiNA,6B;MAjNA,wB;QAEuD,ODMD,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,EAAY,CDiN9C,cEvNwC,KFuN9B,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KCRtD,C;gGAGA,yB;MA8MA,6B;MA9MA,wB;QAEwD,ODGF,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,EAAY,CCKn9C,cArNyC,KAqN/B,KAAL,GAAiB,KAAtB,CDIN8C,MAAZ,CAAf,C;O;KCLtD,C;gGAGA,yB;MA2MA,6B;MA3MA,wB;QAEsD,ODAA,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,ECAK,KDAO,KAAZ,CAAf,C;O;KCFtD,C;gGAGA,yB;MAkNA,aAS6D,0B;MAT7D,+B;MAiNA,wB;QAEwD,OeAA,eAAW,CfyN9B,eAAW,oBAAL,SAAK,CAAL,YAAN,CeZn8B,MAAK,UfAK,KeAO,KAAZ,CAAhB,C;O;KfFxD,C;4FAIA,yB;MAoMA,6B;MD9LA,4C;M CNA,wB;QAEqD,ODMD,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,EDoMjB,cE1MoC,KF0M1B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KCRpD,C;4FAGA,yB;MAiMA,6B;MD9LA,4C;MCHA,wB;QAEsD,ODGF,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAxMqC,KAwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KCLpD,C;4FAGA,yB;MA8LA,6B;MD9LA,4C;MCAA,wB;QAEoD,ODAA,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECakB,KDAIB,C;O;KCFpD,C;4FAGA,yB;MAqMA,aAS6D,0B;MAT7D,+B;MerMA,8C;MfAA,wB;QAEsD,OeAA,Yf4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce5MiB,EfAmB,KeAnB,C;O;KfFtD,C;4FAIA,yB;MAuLA,6B;MDzKA,kD;MCdA,wB;QAMqD,ODcD,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cEzLoC,KFyL1B,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KCPbD,C;4FAOA,yB;MAgLA,6B;MDzKA,kD;MCPA,wB;QAMsD,ODOF,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4KjB,cAnLqC,KAmL3B,KAAL,GAAiB,KAAtB,CD5KiB,C;O;KCbpD,C;4FAOA,yB;MAyKA,6B;MDzKA,kD;MCAA,wB;QAMoD,ODAA,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ECakB,KDAIB,C;O;KCNpD,C;4FAOA,yB;MA4KA,aAS6D,0B;MAT7D,+B;Me5KA,oD;MfAA,wB;QAMsD,OeAA,ef+KjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce/KiB,EfAmB,KeAnB,C;O;KfNtD,C;sGAQA,yB;MA0JA,6B;MD9LA,4C;MCoCA,wB;QAMiD,ODxCG,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,EDoMjB,cE5JqC,KF4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KCkCpD,C;sGAOA,yB;MAmJA,6B;MD9LA,4C;MC2CA,wB;QAMkD,OD/CE,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAtJsC,KAsJ5B,KAAAL,GAAiB,KAAtB,CDrMiB,C;O;KCycpD,C;sGAOA,yB;MA4IA,6B;MD9LA,4C;MCKDA,wB;QAMgD,ODtDI,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECsDmB,KDtDnB,C;O;KCgDpD,C;sGAOA,yB;MA+IA,aAS6D,0B;MAT7D,+B;MerMA,8C;MfsDA,wB;QAMkD,Oe1DI,Yf4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce5MiB,Ef0DoB,Ke1DpB,C;O;KfoDtD,C;4FAQA,yB;MA6HA,6B;MDzKA,kD;MDuOJ,0B;MAAA,+B;ME3LI,wB;QAQ6C,OF8LR,eAAW,OC5OI,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cE7H4B,KF6HIB,KAAL,GAAiB,GAAtB,CC3KiB,CakLf,KD0DW,CAAX,C;O;KETMrC,C;4FASA,yB;MAoHA,6B;MDzKA,kD;MCwoJ,4B;MAAA,iC;MAnLI,wB;QAQ+C,OAsLR,gBAAY,QD7OC,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4KjB,cArH8B,KAqHpB,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KA9LvC,C;4FASA,yB;MA2GA,6B;MDzKA,kD;MC8DA,wB;QAQ2C,ODhES,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ECgES,KDhET,C;O;KCwDpD,C;4FASA,yB;MA4GA,aAS6D,0B;MAT7D,+B;Me5KA,oD;MfgEA,wB;QAQ6C,OelES,ef+KjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce/KiB,EfkEU,KeIEV,C;O;Kf0DtD,C;4EAUA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;4EAQA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;oGAQA,yB;MAAA,8C;MAwEA,6B;MAxEA,wB;QAE+D,0BA+E5B,cAAU,SAAL,GAAiB,KAAtB,CA/E4B,EA+E5B,cA/EqD,KA+E3C,KAAL,GAAiB,KAAtB,CA/E4B,C;O;KAF/D,C;4FAIA,yB;MAAA,iC;M2LnNJ,4B;M3LmNI,wB;QAEqD,uB2LiNiC,Q3LkN1B,IAAK,K2LINX,G3LkNoB,KAAM,K2LINM,C3LkNjC,C;O;KAFrD,C;0FAGA,yB;MAAA,iC;M2LjNJ,4B;M3LiNI,wB;QAEoD,uB2LhNgC,Q3LgNzB,IAAK,K2LhNX,G3LgNmB,KAAM,K2LhNM,C3LgNhC,C;O;KAFpD,C;4FAGA,yB;MAAA,iC;M2L/MJ,4B;M3L+MI,wB;QAEqD,uB2L9MiC,Q3L8M1B,IAAK,K2L9MX,G3L8MoB,KAAM,K2L9MM,C3L8MjC,C;O;KAFrD,C;4EAGA,yB;MAAA,iC;M2L7MJ,4B;M3L6MI,mB;QAEkC,uB2L5MsB,QAAP,C3L4MR,S2L5Me,C3L4MtB,C;O;KAFIC,C;kFAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,C;O;KAVxC,C;oFAWA,

Y;MASqC,gB;K;gFACrC,Y;MASiC,OAAK,SAAL,GAAiB,K;K;kFACID,yB;MAAA,aASqD,0B;MATrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,Y;O;KATnC,C;oFAWA,yB;MF+DJ,0B;MAAA,+B;ME/DI,mB;QASqC,OFiEE,eAAW,OEjEb,SFiEa,CAAX,C;O;KE1EvC,C;sFAUA,Y;MAEuC,W;K;kFACvC,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,KAAtB,C;O;KATnC,C;oFAUA,yB;MAAA,aAS6D,0B;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,YAAN,C;O;KATrC,C;oFAWA,Y;MAMqC,OApDC,SAAL,GAAiB,K;K;sFAqDI D,Y;MAMuC,OA3DD,SAAL,GAAiB,K;K;gCA6DID,Y;MAAyC,OAAQ,CA7DX,SAAL,GAAiB,KA6DD,Y;K;gCA3UrD,Y;MAAA,c;MAG6D,qD;MAH7D,a;K;8BAAA,iB;MAAA,2IAG6D,oCAH7D,G;K;0EA+UA,yB;MAAA,iC;MAAA,4B;QAW4C,uBAAY,SAAZ,C;O;KAX5C,C;4EAYA,yB;MAAA,iC;MAAA,4B;QAU6C,uBAAO,SAAP,C;O;KAV7C,C;4EAWA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW2C,uBAAY,QAAL,SAAK,CAAZ,C;O;KAX3C,C;4EAYA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW4C,uBAAY,QAAL,SAAK,SAAZ,C;O;KAX5C,C;IiC/WA,8B;MACqB,sB;K;wCAKjB,iB;MAM8C,OjCsVL,WiCtVK,aAAQ,KAAR,CjCsVL,C;K;wCiCpVzC,wB;MAOI,aaAQ,KAAR,IAAiB,KjC4OgB,K;K;mFiCxOP,Y;MAAQ,OAAA,YAAQ,O;K;qCAE9C,Y;MAC+E,gCAAS,YAAT,C;K;IAGzD,qC;MAAkC,yB;MAAjC,oB;MACnB,eAAoB,C;K;6CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;gDACvC,Y;MAA0D,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjCgUS,WiChUe,aAAM,mBAAN,EAAM,2BAAN,OjCgUf,C;;QiChU8C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;2CAG7F,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,K;MAEzC,OAAe,WAAR,YAAQ,EAAS,OjCsNS,KiCtNIB,C;K;gDAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QjB0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aaAO,I;UAAp,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UiB1nD6B,2BjB0nDR,OiB1nDQ,S;UAAA,W;YAAwB,oBAAR,YAAQ,EjB0nDhC,OhBx6CA,KiCInG,C;;UjB0nD/C,IAAI,OAAJ,C;YAAyB,aaAO,K;YAAP,e;;;QAC/C,aaAO,I;;;MiB3nDH,iB;K;oCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,uC;MAAA,qD;MACgC,wBAAK,eAAW,IAAX,CAAL,C;MADhC,Y;K;;;qCAPJ,Y;MAAA,OAKqB,sDALrB,M;K;qCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;mCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;kFAyDA,yB;MAAA,2C;MAWwC,0C;QAAA,wB;UAAW,OAAA,aaAK,KAAL,CjCiMv,K;S;O;MiC5MzC,6B;QAWI,OAAO,qBAAY,gCAAW,IAAX,GAAiB,wBAAjB,CAAZ,C;O;KAXX,C;oFAcA,oB;MAGwE,e;K;IiM5ExE,sC;MAQ2D,OAAa,WAAb,SpOwQjB,KAAL,GAAiB,GoOxQkB,EAAS,KAAT,C;K;IAExE,sC;MAQ4D,OAAa,WAAb,SIO+PIB,KAAL,GAAiB,KkO/PmB,EAAS,KAAT,C;K;IAGzE,sC;MAQ0D,OAAc,WnOiR5B,oBmOjRc,SnOiRnB,KAAL,CAAL,iBmOjRiC,EAAS,KAAT,C;K;IAExE,sC;MAOgD,uBAAc,SnNyQvB,KmNzQS,EAA6B,WAAW,KAAX,CAA7B,C;K;IAGhD,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAGvE,+B;MAMuC,Q;MAAA,2DAAoB,kBAAkB,SAAlB,C;K;IAE3D,sC;MAOiD,Q;MAAA,2CAAE,KAaf,oBAAYB,kBAAkB,SAAlB,C;K;IAE1E,6B;MAMmC,Q;MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IAErD,oC;MAO6C,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAEpE,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAMvE,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASmB,Q;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,InO/EKE,YmO+E9D,GnO/E+E,KAAjB,EAA6B,CD6P5D,SoO9KzB,6BAAm,UpO8K6B,KAAL,GAAiB,GAAtB,CC7P4D,MAA7B,CmO+E9D,IAAJ,C;QAA2B,OAAO,I;MACIC,OpO8OqC,UAAW,OoO9OzC,GnOoL8B,KD0DW,CAAX,C;K;IoO3OzC,mC;MAM8C,mCAAuB,EAAvB,C;K;IAE9C,4C;MASmB,Q;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,InOrGkE,YmOqG9D,GnOrG+E,KAAjB,EAA6B,CC8P5D,SkOzJzB,8BAAO,UIOyJ4B,KAAL,GAAiB,KAAtB,CD9P4D,MAA7B,CmOqG9D,IAAJ,C;QAA4B,OAAO,I;MACnC,OIOyNuC,WAAy,QkOzN5C,GnOwKgC,KCiDY,CAAZ,C;K;IkOtN3C,iC;MAM0C,iCAAqB,EAArB,C;K;IAE1C,0C;MASI,WAAW,KAAX,C;MAEA,aaAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAkB,4BAAK,U;MACvB,S;MAEA,gBAAgB,qBAAK,C AAL,C;MACHB,IAAI,YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,QAAQ,C;;MAGZ,uBAAuB,mB;MAEvB,qBAAqB,gB;MACrB,anOuMmC,SmOvMtB,KnOuMsB,C;MmOtMnC,aaAa,W;MACb,aaAU,KAAY,MAAsB,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,InOnJ8D,YmOmJ1D,MnOnJ2E,KAAjB,EmOmJjD,cnOnJ8E,KAA7B,CmOmJ1D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBnO5FwC,WmO4FvB,KnO5FuB,EmO4Ff,MnO5Fe,C;YmO8FxC,InOvJsD,YmOuJlD,MnOvJmE,KAAjB,EmOuJzC,cnOvJsE,KAA7B,CmOuJlD,IAAJ,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIff,SnOnHkD,SAAe,YmOmHjE,MnOnH4D,KAAL



,EmOmHvD,MnOnHmE,KAAZ,CAAf,C;QmOqHID,mBAAmB,M;QACnB,SnOhJiD,SmOgJjD,MnOhJ2D,KAAK,GAAW,CAkU5C,SmOILrB,KnOkLqB,CAIU4C,MAAX,IAAf,C;QmOiJjD,InOnK8D,YmOmK1D,MnOnK2E,KA AjB,EmOmKjD,YnOnK8E,KAA7B,CmOmK1D,IAAJ,C;UAA2B,OAAO,I;;MAGtC,OAAO,M;K;IAGX,kC;MAM 4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASI,WAAW,KAAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAA d,C;QAAiB,OAAO,I;MAExB,YAAmB,6BAAM,U;MACzB,S;MAEA,gBAAGb,qBAAK,CAAL,C;MACHB,IAAI, YAAAY,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER, QAAQ,C;;MAIZ,uBAAuB,gD;MAEvB,qBAAqB,gB;MACrB,anN0IqC,UAAW,oBmN1InC,KnN0ImC,CAAX,C; MmNzIrC,aAAa,2B;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAAY,QAAQ,qBAAK,CAAL,CAAR,EAAi B,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,InN5M+D,amN4M3D,MnN5M6E,KAAIB,Em N4MID,cnN5MgF,KAA9B,CmN4M3D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAlB,QAAJ,C;YACI,iBnN1J0C,YmN0J zB,KnN1JyB,EmN0JjB,MnN1JiB,C;YmN4J1C,InNhNuD,amNgNnD,MnNhNqE,KAAIB,EmNgN1C,cnNhNwE,K AA9B,CmNgNnD,IAAJ,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIf,SnNjLoD,UmNiLpD,MnNjL+D,KAAK,UmNi L1D,MnNjLsE,KAAZ,CAAhB,C;QmNmLpD,mBAAmB,M;QACnB,SnN9MmD,UmN8MnD,MnN9M8D,KAAK, KAAW,ChBsQ7C,UAAW,oBAAL,CayDR,SmOjHrB,KnOiHqB,CAzDQ,MAAK,CAAL,iBAAN,CgBtQ6C,MAA X,CAAhB,C;QmN+MnD,InN5N+D,amN4N3D,MnN5N6E,KAAIB,EmN4NID,YnN5NgF,KAA9B,CmN4N3D,IA AJ,C;UAA2B,OAAO,I;;MAGtC,OAAO,M;K;I3N9RX,6B;MACkD,OAAuB,0BAAtB,KAAO,WAAe,EAAU,KAA O,WAAjB,C;K;IACzE,8B;MACqD,OAAC,gCAAU,iBAAU,gCAAV,C;K;IAE7E,4B;MACoD,ORiZZ,SAvGI,oB Q1SS,ER0Sd,KAAK,CAAL,iBQ1Sc,KR0ST,oBQ1SuB,ER0S5B,KAAK,CAAL,iBQ1Sc,CRiZH,QAAV,C;K;IQhZ xC,+B;MACuD,OR+Yf,SAvGI,oBQxSY,ERwSjB,KAAK,CAAL,iBQxSiB,QRwSZ,oBQxS0B,ERwS/B,KAAK,C AAL,iBQxSiB,CR+YN,QAAV,C;K;IQ1YxC,6B;MAEI,eAAe,EQkSoB,K;MRjSnC,cAAc,EQiSqB,K;MRhSnC,IA AI,qBAAU,CAAd,C;QACI,OQ6C+D,aR7CpD,EQ6CsE,KAAIB,ER7C/C,EQ6C6E,KAA9B,CR7CpD,IAAJ,GAAa, aAAb,GAA2B,a;;MAItC,IAAI,uBAAAY,CAAhB,C;QACI,OAAO,UAAm,aAAW,OAAX,CAAN,C;;MAIX,eAAiB, 4BAAc,CAAd,CAAD,KAAoB,OAAPB,CAAD,WAAkC,CAAIC,C;MACf,UAAU,kBAAW,kBAAW,OAAX,CAA X,C;MACV,OAAO,UAAm,iCQkCsD,aAAkB,CRICzD,UAAm,GAAN,CQkCyD,MAAIB,EAA8B,CRICvD,UAA M,OAAN,CQkCuD,MAA9B,CRICvC,KAAJ,GAAkC,CAAIC,GAAyC,CAAPD,EAAN,C;K;IAIX,gC;MAKe,Q;M AHX,eAAe,EQ8QoB,K;MR7QnC,cAAc,EQ6QqB,K;MR5QnC,IAAI,qBAAU,CAAd,C;QACW,IQyBwD,aRzBpD, EQyBsE,KAAIB,ERzB/C,EQyB6E,KAA9B,CRzBpD,IAAJ,C;UACH,S;;UAEA,OQgDgD,URhDhD,EQgD2D,KA AK,URhD3D,EQgDuE,KAAZ,CAAhB,C;;QRnDpD,W;;MAQJ,IAAI,uBAAAY,CAAhB,C;QACI,OAAO,UAAm,gB AAW,OAAX,CAAN,C;;MAIX,eAAiB,4BAAc,CAAd,CAAD,KAAoB,OAAPB,CAAD,WAAkC,CAAIC,C;MACf, UAAU,kBAAW,kBAAW,OAAX,CAAX,C;MACV,OAAO,UAAm,aQUsD,aAAkB,CRV9D,UAAm,GAAN,CQU8 D,MAAIB,EAA8B,CRV5D,UAAm,OAAN,CQU4D,MAA9B,CRV5C,KAAJ,GAAkC,OAIC,KAAN,CAAN,C;K; IAGX,yB;MAEI,IAAE,QAAF,CAAE,CAAF,C;QADyC,OAC5B,W;;QACb,SRwSuC,aQxSiC,4BAAK,URwS0C,K AAb,CQxSvC,C;UAFyC,OAEP,4BAAK,U;;UACvC,SRuSuC,aQvSiC,4BAAK,URuS0C,KAAb,CQvSvC,C;YAHy C,OAGP,4BAAK,U;eACvC,SAAK,UAAAL,C;YAJyC,ORkVN,SQ9UX,YAAF,CAAE,CR8UW,C;;YQIVM,ORgBY ,SAAU,CAkU5B,SQ7UP,YAAmB,IAAI,UAAe,CR6UO,CAIU4B,MAAK,GAAW,CAkU5C,SQ7UY,UR6UZ,CAIU 4C,MAAX,IAAf,C;;;K;IQRzD,0B;MAEI,IAAE,QAAF,CAAE,CAAF,C;QAD2C,OAC9B,2B;;QACb,SQkSuC,cRI SIC,6BAAM,UQkS0C,KAAAd,CRISvC,C;UAF2C,OAER,6BAAM,U;;UACzC,SQiSuC,cRjSiC,6BAAM,UQiS0C,K AAd,CRjSvC,C;YAH2C,OAGR,6BAAM,U;eACzC,4C;YAJ2C,OQwVL,URpVd,uBAAF,CAAE,CQoVc.C;;YRxV K,OQUY,UAAW,CA8U5B,URjVF,uBAA3B,IAAI,oBAAuB,CQIVe,CA9U4B,MAAK,KAAW,CRHzB,gCQGyB, MAAX,CAAhB,C;;;K;IRC3D,yB;MAC4C,QAAC,CAAqB,GAaf,UAAP,IAAmC,CAAC,MAAO,EAAW,IAAJ,E AAf,IAAgC,C;K;IAE/G,0B;MAC8C,OAAC,qBAAO,EAAP,CAAW,WAAZ,GAAyB,IAAZB,GAAiC,YAAjC,W;K ;IAG9C,0B;MAA8C,uBAAc,CAAd,EAAiB,EAAjB,C;K;IAE9C,kC;MACI,IAAI,gBAAK,CAAT,C;QAAY,OAAS, WAAF,CAAE,EAAS,IAAT,C;MAErB,eAAiB,qBAAO,CAAP,CAAD,yBAAa,IAAb,EAAD,WAAwB,CAAxB,C; MACf,UAAU,WAAI,sCAAW,IAAX,EAAJ,C;MACV,IAAI,kBAAO,IAAX,C;QACI,uCAAo,IAAP,E;QACA,4CA AY,CAAZ,E;;MAEJ,OAAgB,WAAT,QAAS,EAAS,IAAT,CAAT,GAA8B,WAAJ,GAAL,EAAS,IAAT,C;K;I4N1F zC,qC;K;.....



direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.76 libpng 1.6.38

## 1.76.1 Available under license :

---

Copyright (c) 1998-2008 Greg Roelofs. All rights reserved.

This software is provided "as is," without warranty of any kind, express or implied. In no event shall the author or contributors be held liable for any damages arising in any way from the use of this software.

The contents of this file are DUAL-LICENSED. You may modify and/or redistribute this software according to the terms of one of the following two licenses (at your option):

LICENSE 1 ("BSD-like with advertising clause"):

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, disclaimer, and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by Greg Roelofs and contributors for the book, "PNG: The Definitive Guide," published by O'Reilly and Associates.

LICENSE 2 (GNU GPL v2 or later):

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software Foundation,  
Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

-----  
pnm2png / png2pnm --- conversion from PBM/PGM/PPM-file to PNG-file

copyright (C) 1999-2019 by Willem van Schaik <willem at schaik dot com>

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or  
implied, including but not limited to the warranties of merchantability,  
fitness for a particular purpose and noninfringement. In no event shall the  
authors or copyright holders be liable for any claim, damages or other  
liability,  
whether in an action of contract, tort or otherwise, arising from,  
out of or in connection with the software or the use or other dealings in the  
software.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This  
General Public License applies to most of the Free Software  
Foundation's software and to any other program whose authors commit to



using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide

a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions

either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least

the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the  
GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
```

```
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

=====

PNG Reference Library License version 2

-----

- \* Copyright (c) 1995-2022 The PNG Reference Library Authors.
- \* Copyright (c) 2018-2022 Cosmin Truta.
- \* Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
- \* Copyright (c) 1996-1997 Andreas Dilger.
- \* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

-----

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are



Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Mans Rullgard  
Cosmin Truta  
Gilles Vollant  
James Yu  
Mandar Sahastrabudde  
Google Inc.  
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey

Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

Some files in the "scripts" directory have other copyright owners,  
but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are  
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"  
is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing  
Authors and Group 42, Inc. disclaim all warranties, expressed or  
implied, including, without limitation, the warranties of  
merchantability and of fitness for any purpose. The Contributing  
Authors and Group 42, Inc. assume no liability for direct, indirect,  
incidental, special, exemplary, or consequential damages, which may  
result from the use of the PNG Reference Library, even if advised of  
the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this  
source code, or portions hereof, for any purpose, without fee, subject  
to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not  
be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from  
any  
source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit,  
without fee, and encourage the use of this source code as a component  
to supporting the PNG file format in commercial products. If you use  
this source code in a product, acknowledgment is not required but would  
be appreciated.

# 1.77 zlib 1.2.13-r1

## 1.77.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly      Mark Adler  
jloup@gzip.org      madler@alumni.caltech.edu

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE

FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.78 jctools-core 4.0.1

### 1.78.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0  
Bnd-LastModified: 1662622707903  
Build-Jdk-Spec: 11  
Bundle-Description: Java Concurrency Tools Core Library  
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>  
Bundle-ManifestVersion: 2  
Bundle-Name: Java Concurrency Tools Core Library  
Bundle-SymbolicName: org.jctools.core  
Bundle-Version: 4.0.1  
Created-By: Apache Maven Bundle Plugin 5.1.6  
Export-Package: org.jctools.maps;version="4.0.1",org.jctools.util;uses:="sun.misc";version="4.0.1",org.jctools.queues;version="4.0.1",org.jctools.queues.atomic;uses:="org.jctools.queues";version="4.0.1",org.jctools.queues.unpadded;uses:="org.jctools.queues";version="4.0.1",org.jctools.counters;version="4.0.1"  
Import-Package: sun.misc;resolution:=optional  
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.6))"  
Tool: Bnd-6.2.0.202202251641

Found in path(s):

\* /opt/cola/permits/1500645298\_1670406227.9308958/0/jctools-core-4-0-1-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

\* /opt/cola/permits/1500645298\_1670406227.9308958/0/jctools-core-4-0-1-jar/META-INF/maven/org.jctools/jctools-core/pom.xml

## 1.79 scala 2.12.10.v20190904-150159-VFINAL-61701c2

## 1.79.1 Available under license :

Scala includes the JLine library:

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the ASM library.

Copyright (c) 2000-2011 INRIA, France Telecom  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided



that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Scala

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

Scala includes software developed at  
LAMP/EPFL (<https://lamp.epfl.ch/>) and  
Lightbend, Inc. (<https://www.lightbend.com/>).

Licensed under the Apache License, Version 2.0 (the "License").  
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

This software includes projects with other licenses -- see `doc/LICENSE.md`.  
not-a-legal-formal-parameter-tuple.scala:2: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

Either create a single parameter accepting the Tuple2,  
or consider a pattern matching anonymous function: `{ case (a, b) => ... }`  
val x: ((Int, Int) => Int) = (((a, b)) => a)  
^

not-a-legal-formal-parameter-tuple.scala:3: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

Either create a single parameter accepting the Tuple2,  
or consider a pattern matching anonymous function: `{ case (param1, param2) => ... }`  
val y: ((Int, Int, Int) => Int) = (((a, !)) => a)  
^

not-a-legal-formal-parameter-tuple.scala:4: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

Either create a single parameter accepting the  
Tuple3,  
or consider a pattern matching anonymous function: `{ case (param1, ..., param3) => ... }`  
val z: ((Int, Int, Int) => Int) = (((a, NotAPatternVariableName, c)) => a)  
^

three errors found

```
{% if site.thisScalaVersion != site.latestScalaVersion %}
<div class="version-notice">This is the specification of a previous version of Scala. See the <a href="{ {
site.baseurl} }"/>{ { site.latestScalaVersion } }"/>Scala { { site.latestScalaVersion } } spec.</div>
{% endif %}
```

(The MIT License)

Copyright (c) 2013 Greg Allen

Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated documentation files (the  
'Software'), to deal in the Software without restriction, including  
without limitation the rights to use, copy, modify, merge, publish,  
distribute, sublicense, and/or sell copies of the Software, and to  
permit persons to whom the Software is furnished to do so, subject to  
the following conditions:

The above copyright notice and this permission notice shall be  
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY  
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE  
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Scala is licensed under the [Apache License Version 2.0](https://www.apache.org/licenses/LICENSE-2.0).

## Scala License

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# Other Licenses

This software includes projects with the following licenses,  
which are also included in the `licenses/`` directory:

### [Apache License](http://www.apache.org/licenses/LICENSE-2.0.html)

This license is used by the following third-party libraries:

- \* jansi

### [BSD License](http://www.opensource.org/licenses/bsd-license.php)

This license is used by the following third-party libraries:

- \* jline

### [BSD 3-Clause License](http://opensource.org/licenses/BSD-3-Clause)

This license is used by the following third-party libraries:

- \* asm

### [MIT License](http://www.opensource.org/licenses/MIT)

This license is used by the following third-party libraries:

- \* jquery
- \* tools tooltip

Copyright (c) 2006, Ivan Sagalaev

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the Tools Tooltip library:

Copyright (c) 2009 Tero Piirainen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(c) 2012-2014 GitHub

When using the GitHub logos, be sure to follow the GitHub logo guidelines (<https://github.com/logos>)

Font License: SIL OFL 1.1 (<http://scripts.sil.org/OFL>)

Applies to all font files

Code License: MIT (<http://choosealicense.com/licenses/mit/>)

Applies to all other files

Scala includes the jQuery library:

Copyright (c) 2010 John Resig

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Scala includes the JLine library, which includes the Jansi library.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and



may  
provide additional or different license terms and conditions  
for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade  
names, trademarks, service marks, or product names of the Licensor,  
except as required for reasonable and customary

use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or  
agreed to in writing, Licensor provides the Work (and each  
Contributor provides its Contributions) on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
implied, including, without limitation, any warranties or conditions  
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A  
PARTICULAR PURPOSE. You are solely responsible for determining the  
appropriateness of using or redistributing the Work and assume any  
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,  
whether in tort (including negligence), contract, or otherwise,  
unless required by applicable law (such as deliberate and grossly  
negligent acts) or agreed to in writing, shall

any Contributor be

liable to You for damages, including any direct, indirect, special,  
incidental, or consequential damages of any character arising as a  
result of this License or out of the use or inability to use the  
Work (including but not limited to damages for loss of goodwill,  
work stoppage, computer failure or malfunction, or any and all  
other commercial damages or losses), even if such Contributor  
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing  
the Work or Derivative Works thereof, You may choose to offer,  
and charge a fee for, acceptance of support, warranty, indemnity,  
or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the  
License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.80 slf4j 1.6.6

## 1.80.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*\*

\* Copyright (c) 2004-2011 QOS.ch

\* All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without limitation the rights to use, copy, modify, merge, publish,  
 \* distribute, sublicense, and/or sell copies of the Software, and to  
 \* permit persons to whom the Software is furnished to do so, subject to  
 \* the following conditions:  
 \*  
 \* The above copyright notice and this permission notice shall be  
 \* included in all copies or substantial portions of the Software.  
 \*  
 \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
 \* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
 \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
 \* NONINFRINGEMENT. IN  
 NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE  
 \* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION  
 \* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
 \* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
 \*  
 \*/

Found in path(s):

\* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/IMarkerFactory.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/BasicMDCAdapter.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/MarkerIgnoringBase.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/spi/MDCAdapter.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/SubstituteLoggerFactory.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/LoggerFactory.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/impl/StaticLoggerBinder.java  
 \*  
 /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/NOPMDCAdapter.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/MarkerFactory.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/impl/StaticMDCBinder.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/BasicMarkerFactory.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/spi/LocationAwareLogger.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/FormattingTuple.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/spi/MarkerFactoryBinder.java  
 \* /opt/cola/permits/1512417657\_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/MessageFormatter.java  
 \*

```

/opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/helpers/NamedLoggerBase.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/spi/LoggerFactoryBinder.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/MDC.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/helpers/Util.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/helpers/NOPLLogger.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/ILoggerFactory.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/Logger.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/helpers/NOPLLoggerFactory.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/helpers/BasicMarker.java
*
/opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-jar/org/slf4j/Marker.java
* /opt/cola/permits/1512417657_1673636624.6005318/0/slf4j-api-1-6-6-sources-
jar/org/slf4j/impl/StaticMarkerBinder.java

```

# 1.81 protobuf-java-util 3.21.10

## 1.81.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

```

Found in path(s):

```

* /opt/cola/permits/1518087455_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-
jar/com/google/protobuf/util/FieldMaskUtil.java
* /opt/cola/permits/1518087455_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-
jar/com/google/protobuf/util/JsonFormat.java
*
/opt/cola/permits/1518087455_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-
jar/com/google/protobuf/util/FieldMaskTree.java
* /opt/cola/permits/1518087455_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-
jar/com/google/protobuf/util/Values.java
* /opt/cola/permits/1518087455_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-
jar/com/google/protobuf/util/Structs.java

```

\* /opt/cola/permits/1518087455\_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-jar/com/google/protobuf/util/Timestamps.java  
\* /opt/cola/permits/1518087455\_1671610010.5919955/0/protobuf-java-util-3-21-10-sources-jar/com/google/protobuf/util/Durations.java

# 1.82 jackson 2.14.2

## 1.82.1 Available under license :

FasterXML, LLC

Software Grant and Corporate Contributor License Agreement ("Agreement")

<https://github.com/FasterXML/jackson/blob/master/contributor-agreement-corporate.txt>

(v1)

Thank you for your interest in FasterXML, LLC ("FasterXML"). In order to clarify the intellectual property license granted with Contributions from any person or entity, FasterXML must have a Contributor License Agreement (CLA) on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of FasterXML and its users; it does not change your rights to use your own Contributions for any other purpose.

This version of the Agreement allows an entity (the "Corporation") to submit Contributions to the FasterXML, to authorize Contributions submitted by its designated employees to FasterXML, and to grant copyright and patent licenses thereto.

If you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to [clas@fasterxml.com](mailto:clas@fasterxml.com). If necessary, send an original signed Agreement to FasterXML, LLC, 600 N 36th Ave, Suite 409, Seattle, WA 98103.

Please read this document carefully before signing and keep a copy for your records.

Corporation name: \_\_\_\_\_

Corporation address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Point of Contact: \_\_\_\_\_

E-Mail: \_\_\_\_\_

Telephone: \_\_\_\_\_ Fax: \_\_\_\_\_

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to FasterXML. Except for the license granted herein to FasterXML and recipients of software distributed by FasterXML, You reserve all right, title, and interest in and to Your Contributions.

#### 1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with FasterXML. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean the code, documentation or other original works of authorship expressly identified in Schedule B, as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to FasterXML for inclusion in, or documentation of, any of the products owned or managed by FasterXML (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to FasterXML or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, FasterXML for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to recipients of software distributed by FasterXML a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to recipients of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) were submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.
4. You represent that You are legally entitled to grant the above license. You represent further that each employee of the Corporation designated on Schedule A below (or in a subsequent written modification to that Schedule) is authorized to submit Contributions on behalf of the Corporation.
5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others).
6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.
7. Should You wish to submit work that is not Your original creation, You may submit it to FasterXML separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".

8. It is your responsibility to notify FasterXML when any change is required to the list of designated employees authorized to submit Contributions on behalf of the Corporation, or to the Corporation's Point of Contact with FasterXML.

Please sign: \_\_\_\_\_ Date: \_\_\_\_\_

Title: \_\_\_\_\_

Corporation: \_\_\_\_\_

#### Schedule A

[Initial list of designated employees. NB: authorization is not tied to particular Contributions.]

#### Schedule B

[Identification of optional concurrent software grant. Would be left blank or omitted if there is no concurrent software grant.]

## 1.83 jackson-annotations 2.14.2

### 1.83.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.



"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Maven Wrapper Jar  
Copyright 2016-2021 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.  
It was originally written by Tatu Saloranta ([tatu.saloranta@iki.fi](mailto:tatu.saloranta@iki.fi)), and has  
been in development since 2007.  
It is currently developed by a community of developers.

## ## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0  
To find the details that apply to this artifact see the accompanying LICENSE file.

## ## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included  
in some artifacts (usually source distributions); but is always available  
from the source code management (SCM) system project uses.

# 1.84 jackson-databind 2.14.2

## 1.84.1 Available under license :

FasterXML, LLC

Software Grant and Corporate Contributor License Agreement ("Agreement")

<https://github.com/FasterXML/jackson/blob/master/contributor-agreement-corporate.txt>

(v1)

Thank you for your interest in FasterXML, LLC ("FasterXML"). In  
order to clarify the intellectual property license granted with  
Contributions from any person or entity, FasterXML must have a  
Contributor License Agreement (CLA) on file that has been signed by  
each Contributor, indicating agreement to the license terms  
below. This license is for your protection as a Contributor as well  
as the protection of FasterXML and its users; it does not change  
your rights to use your own Contributions for any other purpose.

This version of the Agreement allows an entity (the "Corporation") to submit Contributions to the FasterXML, to authorize Contributions submitted by its designated employees to FasterXML, and to grant copyright and patent licenses thereto.

If

you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to [clas@fasterxml.com](mailto:clas@fasterxml.com). If necessary, send an original signed Agreement to FasterXML, LLC, 600 N 36th Ave, Suite 409, Seattle, WA 98103.

Please read this document carefully before signing and keep a copy for your records.

Corporation name: \_\_\_\_\_

Corporation address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Point of Contact: \_\_\_\_\_

E-Mail: \_\_\_\_\_

Telephone: \_\_\_\_\_ Fax: \_\_\_\_\_

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to FasterXML. Except for the license granted herein to FasterXML and recipients of software distributed by FasterXML, You reserve all right, title, and interest in and to Your Contributions.

#### 1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with FasterXML. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean the code, documentation or other original works of authorship expressly identified in Schedule B, as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to FasterXML for inclusion in, or documentation of, any of the products owned or managed by FasterXML (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to FasterXML or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, FasterXML for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to

recipients of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to recipients of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s)

were submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that You are legally entitled to grant the above license. You represent further that each employee of the Corporation designated on Schedule A below (or in a subsequent

written modification to that Schedule) is authorized to submit Contributions on behalf of the Corporation.

5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others).
6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.
7. Should You wish to submit work that is not Your original creation, You may submit it to FasterXML separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".
8. It is your responsibility to notify FasterXML when any change is required to the list of designated employees authorized to submit Contributions on behalf of the Corporation, or to the Corporation's Point of Contact with FasterXML.

Please sign: \_\_\_\_\_ Date: \_\_\_\_\_

Title: \_\_\_\_\_

Corporation: \_\_\_\_\_

#### Schedule A

[Initial list of designated employees. NB: authorization is not tied to particular Contributions.]

#### Schedule B



[Identification of optional concurrent software grant. Would be left blank or omitted if there is no concurrent software grant.]

# 1.85 wire-compiler 3.7.1

## 1.85.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1551206746_1675341574.9799197/0/wire-compiler-3-7-1-sources-
jar/com/squareup/wire/schema/WireRun.kt
*/opt/cola/permits/1551206746_1675341574.9799197/0/wire-compiler-3-7-1-sources-
jar/com/squareup/wire/schema/Target.kt
*/opt/cola/permits/1551206746_1675341574.9799197/0/wire-compiler-3-7-1-sources-
jar/com/squareup/wire/schema/Root.kt
*/
/opt/cola/permits/1551206746_1675341574.9799197/0/wire-compiler-3-7-1-sources-
jar/com/squareup/wire/schema/ProfileLoader.kt
*/opt/cola/permits/1551206746_1675341574.9799197/0/wire-compiler-3-7-1-sources-
jar/com/squareup/wire/schema/SchemaLoader.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *

```

- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.

\*/

Found in path(s):

- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/WireLogger.kt
- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/ConsoleWireLogger.kt
- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/WireCompiler.kt

\*

/opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/WireException.kt

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2020 Square Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/schema/PartitionedSchema.kt
- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/Manifest.kt
- \* /opt/cola/permits/1551206746\_1675341574.9799197/0/wire-compiler-3-7-1-sources-jar/com/squareup/wire/schema/DirectedAcyclicGraph.kt

## 1.86 glue 2.17.122

## 1.86.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed

with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.87 schema-registry-serde 1.1.13

### 1.87.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

- \* Copyright 2021 Amazon.com, Inc. or its affiliates.
- \* Licensed under the Apache License, Version 2.0 (the
- \* "License"); you may not use this file except in compliance
- \* with the License. You may obtain a copy of the License at
- \*
- \* <http://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/serializers/protobuf/ProtobufWireFormatEncoder.java
- No license file was found, but licenses were detected in source scan.

/\*

- \* Copyright 2020 Red Hat
- \* Portions Copyright 2020 Amazon.com, Inc. or its affiliates.
- \* All Rights Reserved.
- \*
- \* Licensed under the Apache License, Version 2.0 (the "License");
- \* you may not use this file except in compliance with the License.
- \* You may obtain a copy of the License at
- \*
- \* <http://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.

<<<<<<< HEAD

\*

- \* This will be removed once Apicurio releases the latest version with the json\_name fix



\* <https://github.com/Apicurio/apicurio-registry/blob/master/utils/protobuf-schema-utilities/src/main/java/io/apicurio/registry/utils/protobuf/schema/ProtobufFile.java>

=====  
>>>>>>

kafka-connect-converter-protobuf

\*/

Found in path(s):

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/ProtobufFile.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2021 Red Hat

\* Portions Copyright 2020 Amazon.com, Inc. or its affiliates.

\* All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/ProtobufMessage.java

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/ProtobufSchema.java

\*

/opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/MessageDefinition.java

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/DynamicSchema.java

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/EnumDefinition.java

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/FileDescriptorUtils.java

\* /opt/cola/permits/1551206578\_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/com/amazonaws/services/schemaregistry/utils/apicurio/ProtobufSchemaLoader.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 Amazon.com, Inc. or its affiliates.
 * Licensed under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/json/JsonDeserializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/avro/AvroSerializer.java
*
/opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/SerializationDataEncoder.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/json/JsonDataWithSchema.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/GlueSchemaRegistrySerializerImpl.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/GlueSchemaRegistrySerializerFactory.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryDeserializerFactory.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryDeserializerImpl.java
*
/opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryDeserializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryDeserializerDataParser.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/protobuf/ProtobufWireFormatDecoder.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/json/JsonSerializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/protobuf/ProtobufDeserializer.java
*
/opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/avro/AWSKafkaAvroDeserializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
```

```
jar/com/amazonaws/services/schemaregistry/serializers/json/JsonValidator.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/utils/AVROUtils.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryKafkaDeserializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/avro/AWSKafkaAvroSerializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/GlueSchemaRegistryDeserializationFacade.java
*
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/SecondaryDeserializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/GlueSchemaRegistrySerializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/GlueSchemaRegistrySerializationFacade.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/serializers/protobuf/ProtobufSerializer.java
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/deserializers/avro/AvroDeserializer.java
No license file was found, but licenses were detected in source scan.
```

```
<!--
/*
* Copyright 2020 Amazon.com, Inc. or its affiliates.
* Licensed under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->
```

```
Found in path(s):
* /opt/cola/permits/1551206578_1675341673.5921674/0/schema-registry-serde-1-1-13-sources-jar/META-
INF/maven/software.amazon.glue/schema-registry-serde/pom.xml
```

## 1.88 schema-registry-build-tools 1.1.13

## 1.88.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
/*
* Copyright 2020 Amazon.com, Inc. or its affiliates.
* Licensed under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->
```

Found in path(s):

```
* /opt/cola/permits/1551206651_1675341680.67456/0/schema-registry-build-tools-1-1-13-sources-jar/META-INF/maven/software.amazon.glue/schema-registry-build-tools/pom.xml
```

No license file was found, but licenses were detected in source scan.

```
<!--
/*
* Copyright 2019 Amazon.com, Inc. or its affiliates.
* Licensed under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->
```

Found in path(s):

```
* /opt/cola/permits/1551206651_1675341680.67456/0/schema-registry-build-tools-1-1-13-sources-jar/suppressions.xml
```

```
* /opt/cola/permits/1551206651_1675341680.67456/0/schema-registry-build-tools-1-1-13-sources-jar/checkstyle.xml
```

# 1.89 sts 2.17.122

## 1.89.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](https://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.



8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.90 arns 2.17.122

## 1.90.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding  
those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade  
names, trademarks, service marks, or product names of the Licensor,  
except as required for reasonable and customary use in describing the  
origin of the Work and reproducing the  
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or  
agreed to in writing, Licensor provides the Work (and each  
Contributor provides its Contributions) on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.91 kotlin-scripting-compiler-embeddable

### 1.7.10

#### 1.91.1 Available under license :

Apache-2.0

## 1.92 wire-kotlin-generator 3.7.1

#### 1.92.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1551206675_1675341672.2420092/0/wire-kotlin-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/KotlinGenerator.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1551206675_1675341672.2420092/0/wire-kotlin-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/FieldExtensions.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1551206675_1675341672.2420092/0/wire-kotlin-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/RpcRole.kt
```

```
* /opt/cola/permits/1551206675_1675341672.2420092/0/wire-kotlin-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/RpcCallStyle.kt
```

## 1.93 schema-registry-common 1.1.13



## 1.93.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 Amazon.com, Inc. or its affiliates.
 * Licensed under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/GlueSchemaRegistryCompressionHandler.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSSchemaRegistryClient.java
*
/opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/utils/AvroRecordType.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSSchemaNamingStrategy.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSSchemaNamingStrategyDefaultImpl.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/GlueSchemaRegistryDataFormatDeserializer.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/exception/AWSSchemaRegistryException.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSSchemaRegistryGlueClientRetryPolicyHelper.java
*
/opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSSerializerInput.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/utils/GlueSchemaRegistryUtils.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/GlueSchemaRegistryDefaultCompression.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/utils/AWSSchemaRegistryConstants.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/GlueSchemaRegistryCompressionFactory.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
```

```
jar/com/amazonaws/services/schemaregistry/caching/GlueSchemaRegistryCache.java
*
/opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/GlueSchemaRegistryDataFormatSerializer.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/AWSDeserializerInput.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/exception/GlueSchemaRegistryIncompatibleDataException.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/Schema.java
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-
jar/com/amazonaws/services/schemaregistry/common/configs/GlueSchemaRegistryConfiguration.java
No license file was found, but licenses were detected in source scan.
```

```
<!--
/*
* Copyright 2020 Amazon.com, Inc. or its affiliates.
* Licensed under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->
```

Found in path(s):

```
* /opt/cola/permits/1551206723_1675341578.6784365/0/schema-registry-common-1-1-13-sources-jar/META-
INF/maven/software.amazon.glue/schema-registry-common/pom.xml
```

## 1.94 jackson-dataformat-cbor 2.14.2

### 1.94.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JacksonIon

Copyright 2012-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a



cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.95 kotlin 1.8.10-release-430

## 1.95.1 Available under license :

Copyright JS Foundation and other contributors, <https://js.foundation>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/qunitjs/qunit>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is

modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of

free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER  
GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from

such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility

in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the



source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the

copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license

restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this

License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library

specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting

redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
MIT License

Copyright (c) 2018 Chad Retz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/\*\*

\* Copyright 2010 Tim Down.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
THE SOFTWARE.

Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.

Developed at SunSoft, a Sun Microsystems, Inc. business.

Permission to use, copy, modify, and distribute this  
software is freely granted, provided that this notice  
is preserved.

/\*

\* Copyright 2000-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

\* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

\*/

package tasks

import groovy.util.Node

import groovy.xml.XmlParser

import org.gradle.api.DefaultTask

import org.gradle.api.file.RegularFileProperty

import org.gradle.api.provider.Property

import org.gradle.api.tasks.Input

import org.gradle.api.tasks.InputFile

import org.gradle.api.tasks.OutputFile

import org.gradle.api.tasks.TaskAction

import java.util.\*

abstract class WriteCopyrightToFile : DefaultTask() {

    @InputFile

    val path = project.file("\${project.rootDir}/.idea/copyright/apache.xml")

    @get:OutputFile

    abstract val outputFile: RegularFileProperty

    @get:Input

    val commented: Property<Boolean> = project.objects.property(Boolean::class.java).convention(true)

    @TaskAction

    fun write() {

        val file = outputFile.asFile.get()

        file.writeText(if

(commented.get()) readCopyrightCommented() else readCopyright())

    }

    private fun readCopyright(): String {

        assert(path.exists()) {

            "File \$path with copyright not found"

        }



```

val xmlParser = XmlParser()
val node = xmlParser.parse(path)
assert(node.attribute("name") == "CopyrightManager") {
 "Format changed occasionally?"
}

val copyrightBlock = node.children().filterIsInstance<Node>().single()
val noticeNode = copyrightBlock.children().filterIsInstance<Node>().single { it.attribute("name") == "notice" }
return noticeNode.attribute("value").toString().replace("$today.year",
GregorianCalendar()[Calendar.YEAR].toString())
}

private fun readCopyrightCommented(): String {
 return "/*\n" + readCopyright().prependIndent(" * ") + "\n */"
}
}
/*
* Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
The MIT License

```

Copyright (C) 2011-2019 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR

COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<https://github.com/square/okhttp/>

Copyright 2019 Square, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

zlib.h -- interface of the 'zlib' general purpose compression library  
version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly      Mark Adler  
jloup@gzip.org      madler@alumni.caltech.edu

SCALA LICENSE

Copyright (c) 2002-2012 EPFL, Lausanne, unless otherwise specified.

All rights reserved.

This software was developed by the Programming Methods Laboratory of the

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

Permission to use, copy, modify, and distribute this software in source or binary form for any purpose with or without fee is hereby granted, provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

#### THIS SOFTWARE

IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2016 Jamie Sherriff

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Apache 2 license (given in full in [LICENSE.txt](LICENSE.txt)) applies to all code in this repository which is copyright

by JetBrains. The following sections of the repository contain third-party code, to which different licenses may apply:

### ## Kotlin Compiler

The following modules contain third-party code and are incorporated into the Kotlin compiler and/or the Kotlin IntelliJ IDEA plugin:

- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/inline/MaxStackSizeAndLocalsCalculator.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/inline/MaxLocalsCalculator.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/FastMethodAnalyzer.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/InstructionLivenessAnalyzer.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/ControlFlowGraph.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/fixStack/FastStackAnalyzer.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/temporaryVals/FastStoreLoadAnalyzer.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-

2011 INRIA, France Telecom

- Path: eval4j/src/org/jetbrains/eval4j/interpreterLoop.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
  
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/OptimizationBasicInterpreter.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
  
- Path: js/js.ast
  - License: BSD ([license/third\_party/dart\_LICENSE.txt][dart])
  - Origin: Originally part of the Dart compiler, (c) 2011 the Dart Project Authors,
  
- Path: js/js.inliner/src/org/jetbrains/kotlin/js/inline/FunctionInlineMutator.kt
  - License: BSD ([license/third\_party/dart\_LICENSE.txt][dart])
  - Origin: Originally part of the Dart compiler, (c) 2011 the Dart Project Authors,
  
- Path: js/js.parser/src/com/google
  - License: Netscape Public License 1.1 ([license/third\_party/rhino\_LICENSE.txt][rhino])
  - Origin: Originally part of GWT, (C) 2007-08 Google Inc., distributed under the Apache 2 license. The code is derived from Rhino, (C) 1997-1999 Netscape Communications Corporation, distributed under the Netscape Public License.
  
- Path: js/js.translator/qunit/qunit.js
  - License: MIT([license/third\_party/qunit\_license.txt][qunit])
  - Origin: QUnit, Copyright (c) 2012 John Resig, Jrn Zaeferrer,
  
- Path: libraries/stdlib/src/kotlin/collections
  - License: Apache 2 ([license/third\_party/gwt\_license.txt][gwt])
  - Origin: Derived from GWT, (C) 2007-08 Google Inc.
  
- Path: libraries/stdlib/unsigned/src/kotlin/UnsignedUtils.kt
  - License: Apache 2 ([license/third\_party/guava\_license.txt][guava])
  - Origin: Derived from Guava's UnsignedLongs, (C) 2011 The Guava Authors
  
- Path: libraries/stdlib/jvm/src/kotlin/util/MathJVM.kt
  - License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
  - Origin: Derived from boost special math functions, Copyright Eric Ford & Hubert Holin 2001.
  
- Path: libraries/stdlib/js/src/kotlin/collections
  - License: Apache 2 ([license/third\_party/gwt\_license.txt][gwt])
  - Origin: Derived from GWT, (C) 2007-08 Google Inc.
  
- Path: libraries/stdlib/native-wasm/src/kotlin/collections

- License: Apache 2 ([license/third\_party/gwt\_license.txt][gwt])
- Origin: Derived from GWT, (C) 2007-08 Google Inc.
  
- Path: libraries/stdlib/js-v1/src/js/long.js
- License: Apache 2 ([license/third\_party/closure-compiler\_LICENSE.txt][closure-compiler])
- Origin: Google Closure Library, Copyright 2009 The Closure Library Authors
  
- Path: libraries/stdlib/js-v1/src/js/polyfills.js
- License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
- Origin: Derived from boost special math functions, Copyright Eric Ford & Hubert Holin 2001.
  
- Path: libraries/stdlib/js/src/kotlin/js/math.polyfills.kt
- License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
- Origin: Derived from boost special math functions, Copyright Eric Ford & Hubert Holin 2001.
  
- Path: libraries/stdlib/wasm/internal/kotlin/wasm/internal/Number2String.kt
- License: Apache 2 ([third\_party/assemblyscript\_license.txt][assemblyscript])
- Origin: Derived from assemblyscript standard library
  
- Path: plugins/lint/android-annotations
- License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
- Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: plugins/lint/lint-api
- License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
- Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: plugins/lint/lint-checks
- License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
- Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: plugins/lint/lint-idea
- License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
- Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: wasm/ir/src/org/jetbrains/kotlin/wasm/ir/convertors
- License: MIT ([license/third\_party/asmble\_license.txt][asmble])
- Origin: Copyright (C) 2018 Chad Retz
  
- Path: compiler/tests-common/tests/org/jetbrains/kotlin/codegen/ir/ComposeLikeGenerationExtension.kt
- License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
- Origin: Derived from JetPack Compose compiler plugin code, Copyright 2019 The Android Open Source Project
  
- Path: libraries/stdlib/wasm/src/kotlin/text/FloatingPointConverter.kt
- License: MIT ([license/third\_party/asmble\_license.txt][asmble])
- Origin: Copyright (C) 2018 Chad Retz

- Path: libraries/stdlib/wasm/src/kotlin/math/fdlibm/
- License: SUN ([license/third\_party/sun\_license.txt][sun])
- Origin: Copyright (C) 1993 by Sun Microsystems, Inc.
  
- Path: kotlin-native/runtime/src/main/cpp/Utils.cpp
- License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
- Origin: Derived from boost hash functions, Copyright 2005-2014 Daniel James

## ## Kotlin Test Data

The following source code is used for testing the Kotlin compiler and/or plugin and is not incorporated into any distributions of the compiler, libraries or plugin:

- Path: third-party/annotations/android
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: third-party/annotations/com/android
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: third-party/annotations/org/eclipse
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])
  - Origin: Eclipse JDT, Copyright (c) 2011, 2013 Stephan Herrmann and others.
  
- Path: third-party/annotations/androidx
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: third-party/annotations/edu/umd/cs/findbugs
  - License: LGPL 2.1 ([license/third\_party/testdata/findbugs\_license.txt][findbugs])
  - Origin: Bytecode Analysis Framework, Copyright (C) 2005 University of Maryland
  
- Path: third-party/java8-annotations/org/eclipse
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])
  - Origin: Eclipse JDT, Copyright (c) 2011, 2013 Stephan Herrmann and others.
  
- Path: third-party/annotations/io/reactivex
  - License: Apache 2 ([license/third\_party/testdata/rxjava\_license.txt][rxjava])
  - Origin: RxJava, Copyright (c) 2016-present, RxJava Contributors
  
- Path: third-party/java8-annotations/org/jspecify
  - License: Apache 2 ([license/third\_party/testdata/jspecify\_license.txt][jspecify])
  - Origin: JSpecify, Copyright (C) 2020 The JSpecify Authors
  
- Path: third-party/java9-annotations/org/jspecify
  - License: Apache 2 ([license/third\_party/testdata/jspecify\_license.txt][jspecify])

- Origin: JSpecify, Copyright (C) 2020 The JSpecify Authors
  
- Path: third-party/annotations/lombok
  - License: MIT ([license/third\_party/testdata/lombok\_license.txt][lombok])
  - Origin: Project Lombok, Copyright (C) 2009-2013 The Project Lombok Authors
  
- Path: idea/idea-android/tests/org/jetbrains/kotlin/android/AndroidTestBase.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lintQuickfix/requiresApi/RequiresApi.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lint/IntRange.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lint/RequiresPermission.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-
 tests/src/test/resources/testProject/allOpenSpring/src/org/springframework/stereotype/Component.java
  - License: Apache 2 ([license/third\_party/testdata/spring\_license.txt][spring])
  - Origin: Spring Framework, Copyright 2002-2007 the original author or authors.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/AndroidDaggerProject
  - License: Apache 2 ([license/third\_party/testdata/dagger\_license.txt][dagger])
  - Origin: Dagger, Copyright (C) 2013 Square, Inc.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/kapt2
  - License: Apache 2 ([license/third\_party/testdata/dagger\_license.txt][dagger])
  - Origin: Dagger, Copyright (C) 2013 Square, Inc.
  
- Path: libraries/tools/kotlin-maven-plugin-test/src/it/test-allopen-
 spring/src/main/java/org/springframework/stereotype/Component.java
  - License: Apache
   
2 ([license/third\_party/testdata/spring\_license.txt][spring])
   
- Origin: Spring Framework, Copyright 2002-2007 the original author or authors.

### ## Kotlin Tools and Libraries Tests

The following source code is used for testing the Kotlin tools and/or libraries and is not incorporated into any distributions of the tools or libraries:

- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/kotlin/org/jetbrains/kotlin/gradle/PluginsDslIT.kt



- License: Apache 2 ([license/third\_party/gradle\_license.txt][gradle])
- Origin: Gradle, Copyright 2002-2017 Gradle, Inc.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/target/test-classes/testProject/noArgJpa/src/javax/persistence/Entity.java
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse]) and Eclipse Distribution License - v1.0
  - ([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])
  - Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/noArgJpa/src/javax/persistence/Entity.java
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse]) and Eclipse Distribution License - v1.0
  - ([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])
  - Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/noArgJpa/src/javax/persistence/Embeddable.java
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse]) and Eclipse Distribution License - v1.0
  - ([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])
  - Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.
  
- Path: libraries/tools/kotlin-test-js-runner/karma-kotlin-reporter.js
  - License: MIT ([license/third\_party/karma\_LICENSE.txt](third\_party/karma\_LICENSE.txt) and [license/third\_party/karma-teamcity-reporter\_LICENSE.txt](third\_party/karma-teamcity-reporter\_LICENSE.txt))
  - Origin: Copyright (C) 2011-2019 Google, Inc. and Copyright (C) 2011-2013 Vojta Jna and contributors.
  
- Path: libraries/tools/kotlin-test-js-runner/mocha-kotlin-reporter.js
  - License: MIT ([license/third\_party/mocha-teamcity-reporter\_LICENSE.txt](third\_party/mocha-teamcity-reporter\_LICENSE.txt))
  - Origin: Copyright (c) 2016 Jamie Sherriff
  
- Path: libraries/tools/kotlin-test-js-runner/src/utills.ts
  - License: MIT ([license/third\_party/teamcity-service-messages\_LICENSE.txt](third\_party/teamcity-service-messages\_LICENSE.txt) and [license/third\_party/lodash\_LICENSE.txt](third\_party/lodash\_LICENSE.txt))
  - Origin: Copyright (c) 2013 Aaron Forsander and Copyright JS Foundation and other contributors
  - <<https://js.foundation/>>
  
- Path: libraries/tools/kotlin-test-js-runner/src/teamcity-format.js
  - License: MIT ([license/third\_party/mocha-teamcity-reporter\_LICENSE.txt](third\_party/mocha-teamcity-reporter\_LICENSE.txt) and [license/third\_party/teamcity-service-messages\_LICENSE.txt](third\_party/teamcity-service-

messages\_LICENSE.txt))

- Origin: Copyright (c) 2016 Jamie Sherriff and Copyright (c) 2013 Aaron Forsander

## ## Example Code

The following code is provided as examples and is not incorporated into any distributions of the compiler, libraries or plugin:

- Path: libraries/examples/browser-example/src/js/jquery.js
  - License: MIT ([license/third\_party/jquery\_license.txt][jquery])
  - Origin: jQuery JavaScript Library v1.6.2, Copyright 2011, John Resig
- Path: libraries/examples/browser-example-with-library/src/js/jquery.js
  - License: MIT ([license/third\_party/jquery\_license.txt][jquery])
  - Origin: jQuery JavaScript Library v1.6.2, Copyright 2011, John Resig

[aosp]: third\_party/aosp\_license.txt  
[asm]: third\_party/asm\_license.txt  
[boost]: third\_party/boost\_LICENSE.txt  
[closure-compiler]: third\_party/closure-compiler\_LICENSE.txt  
[dagger]: third\_party/testdata/dagger\_license.txt  
[dart]: third\_party/dart\_LICENSE.txt  
[eclipse]: third\_party/testdata/eclipse\_license.txt  
[eclipse-distribution]: third\_party/testdata/eclipse\_distribution\_license.txt  
[findbugs]: third\_party/testdata/findbugs\_license.txt  
[gradle]: third\_party/gradle\_license.txt  
[guava]: third\_party/guava\_license.txt  
[gwt]: third\_party/gwt\_license.txt  
[jquery]: third\_party/jquery\_license.txt  
[lombok]: third\_party/testdata/lombok\_license.txt  
[qunit]: third\_party/qunit\_license.txt  
[rhino]: third\_party/rhino\_LICENSE.txt  
[rxjava]: third\_party/testdata/rxjava\_license.txt  
[spring]: third\_party/testdata/spring\_license.txt  
[assemblyscript]: third\_party/assemblyscript\_license.txt

ASM: a very small and fast Java bytecode manipulation framework  
Copyright (c) 2000-2005 INRIA, France Telecom  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

---

Licenses for included components:

-----  
Eclipse Public License 1.0  
<https://opensource.org/licenses/EPL-1.0>

junit:junit  
org.sonatype.aether:aether-api  
org.sonatype.aether:aether-connector-wagon  
org.sonatype.aether:aether-impl  
org.sonatype.aether:aether-spi  
org.sonatype.aether:aether-util

-----  
3-Clause BSD  
<https://opensource.org/licenses/BSD-3-Clause>

com.google.code.findbugs:jsr305

org.hamcrest:hamcrest-core  
BSD License

Copyright (c) 2000-2015 [www.hamcrest.org](http://www.hamcrest.org)  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written

permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

com.esotericsoftware.kryo:kryo

com.esotericsoftware.minlog:minlog

Copyright (c) 2008-2018, Nathan Sweet All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Esoteric Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

org.ow2.asm:asm

org.ow2.asm:asm-analysis

org.ow2.asm:asm-commons

org.ow2.asm:asm-tree

org.ow2.asm:asm-util



ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
MIT

com.googlecode.plist:dd-plist

dd-plist - An open source library to parse and generate property lists

Copyright (C) 2016 Daniel Dreibrod

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

org.bouncycastle:bcpg-jdk15on

org.bouncycastle:bcprov-jdk15on

Copyright (c) 2000 - 2019 The Legion of the Bouncy Castle Inc. (<https://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person

obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

org.slf4j:jcl-over-slf4j

org.slf4j:jul-to-slf4j

org.slf4j:log4j-over-slf4j

org.slf4j:slf4j-api

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS

BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
CDDL

<https://opensource.org/licenses/CDDL-1.0>

com.sun.xml.bind:jaxb-impl

-----  
LGPL 2.1

<https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html>

org.samba.jcifs:jcifs

org.jetbrains.intellij.deps:trove4j

-----  
License for the GNU Trove library included by the Kotlin embeddable compiler

-----  
The source code for GNU Trove is licensed under the Lesser GNU Public License (LGPL).

Copyright (c) 2001, Eric D. Friedman All Rights Reserved. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Two classes (HashFunctions and PrimeFinder) included in Trove are licensed under the following terms:

Copyright (c) 1999 CERN - European Organization for Nuclear Research. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

The source code of modified GNU Trove library is available at  
<https://github.com/JetBrains/intellij-deps-trove4j> (with `trove4j_changes.txt` describing the changes)

-----  
Eclipse Distribution License 1.0

<https://www.eclipse.org/org/documents/edl-v10.php>

`org.eclipse.jgit:org.eclipse.jgit`

-----  
BSD-style

`com.jcraft:jsch`

`com.jcraft:jzlib`

Copyright (c) 2000-2011 ymnk, JCraft, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
Eclipse Public License 2.0

<https://www.eclipse.org/legal/epl-2.0/>

`org.junit.platform:junit-platform-launcher`

-----  
Mozilla Public License 2.0  
<https://www.mozilla.org/en-US/MPL/2.0/>

org.mozilla:rhino  
musl as a whole is licensed under the following standard MIT license:

-----  
Copyright 2005-2014 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
Authors/contributors include:

Anthony G. Basile  
Arvid Picciani  
Bobby Bingham  
Boris Brezillon  
Chris Spiegel  
Emil Renner Berthing  
Hiltjo Posthuma  
Isaac Dunham  
Jens Gustedt  
Jeremy Huntwork  
John Spencer  
Justin Cormack  
Luca Barbato  
Luka Perkov

Michael Forney  
Nicholas J. Kain  
orc  
Pascal Cuoq  
Pierre Carrier  
Rich Felker  
Richard Pennington  
Solar Designer  
Strake  
Szabolcs Nagy  
Timo Ters  
Valentin Ochs  
William Haddon

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files).

The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/armel/memcpy.s`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The implementation of DES for crypt (`src/misc/crypt_des.c`) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/misc/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsort.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The BSD PRNG implementation (`src/prng/random.c`) and XSI search API (`src/search/*.c`) functions are Copyright 2011 Szabolcs Nagy and licensed under following terms: "Permission to use, copy, modify, and/or distribute this code for any purpose with or without fee is hereby granted. There is no warranty."

The `x86_64` port was written by Nicholas J. Kain. Several files (`crt`) were released into the public domain; others are licensed under the standard MIT license terms at the top of this file. See individual files for their copyright status.

The `mips` and `microblaze` ports were originally written by Richard Pennington for use in the `elcc` project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration.

It is licensed under the standard MIT terms.

The `powerpc` port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

All public header files (`include/*` and `arch/*/bits/*`) should be treated as Public Domain as they intentionally contain no content which can be covered by copyright. Some source modules may fall in this category as well. If you believe that a file is so trivial that it should be in the Public Domain, please contact the authors and request an explicit statement releasing it from copyright.

The following files are trivial, believed not to be copyrightable in the first place, and hereby explicitly released to the Public Domain:

All public headers: `include/*`, `arch/*/bits/*`

Startup files: `crt/*`

UNLESS SPECIFIED OTHERWISE IN THE INDIVIDUAL SOURCE FILES INCLUDED WITH THIS PACKAGE, they may freely be used, modified and/or redistributed for any purpose.

Portions, Copyright 1991-2005 Unicode, Inc. The following applies to Unicode.

## COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2005 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>. Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

### 2. Additional terms from the Database:

Copyright 1995-1999 Unicode, Inc. All Rights reserved.

#### Disclaimer

The Unicode Character Database is provided as is by Unicode, Inc. No claims are made as to fitness for any particular purpose. No warranties of any kind are expressed or implied. The recipient agrees to determine applicability of information provided. If this file has been purchased on magnetic or optical media from Unicode, Inc., the sole remedy for any claim will be exchange of defective media within 90 days of receipt. This disclaimer is applicable for all other data files accompanying the Unicode Character Database,



some of which have been compiled by the Unicode Consortium, and some of which have been supplied by other sources.

#### Limitations on Rights to Redistribute This Data

Recipient is granted the right to make copies in any form for internal distribution and to freely use the information supplied in the creation of products

supporting the Unicode™ Standard. The files in the Unicode Character Database can be redistributed to third parties or other organizations (whether for profit or not) as long as this notice and the disclaimer notice are retained. Information can be extracted from these files and used in documentation or programs, as long as there is an accompanying notice indicating the source.

Google Dart Js backend was removed - <https://code.google.com/p/dart/source/detail?r=4771>

According to <http://www.apache.org/legal/3party.html> we can include "Google Dart Js backend" in source form, because code license is "New BSD License" (Authorized License).

This part of code will be removed when kotlin will be rewritten on kotlin.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The files in this package are taken from mozilla's Rhino project.

See <http://www.mozilla.org/rhino/>

The files modified from Rhino 1.5R3

(<ftp://ftp.mozilla.org/pub/js/rhino15R3.zip>).

--

The contents of this package are subject to the Netscape Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Rhino code, released May 6, 1999.

The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1997-2000 Netscape Communications Corporation. All Rights Reserved.

Alternatively, the contents of this file may be used under the terms of the GNU Public License (the "GPL"), in which case the provisions of the GPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the GPL and not to allow others to use your version of this file under the NPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL. If you do not delete the provisions above, a recipient may use your version of this file under either the NPL or the GPL.  
<http://fastutil.di.unimi.it/>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares,  
or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2013 Aaron Forsander

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the

Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the



ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must

be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that

uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood

that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to

refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our



decision

will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
Copyright JS Foundation and other contributors, <https://js.foundation/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/jquery>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

The MIT License

Copyright (c) 2003, Kohsuke Kawaguchi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/\*

\* Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.

\* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

\*/

The MIT License

Copyright JS Foundation and other contributors <<https://js.foundation/>>

Based on Underscore.js, copyright Jeremy Ashkenas,

DocumentCloud and Investigative Reporters & Editors <<http://underscorejs.org/>>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/lodash/lodash>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

====

Files located in the node\_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2021, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose

with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings

in this Software without prior written authorization of the copyright holder.

The Trove library is licensed under the Lesser GNU Public License, which is included with the distribution in a file called `trove_license.txt`.

The PrimeFinder and HashFunctions classes in Trove are subject to the following license restrictions:

Copyright (c) 1999 CERN - European Organization for Nuclear Research.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

The version of Rhino used in GWT is licensed under a dual license, Netscape Public License 1.1 / GNU General Public License.

The text of the Netscape Public License

is provided below (<http://website-archive.mozilla.org/www.mozilla.org/mpl/MPL/NPL/1.1/>):

## AMENDMENTS

The Netscape Public License Version 1.1 ("NPL") consists of the Mozilla Public License Version 1.1 with the following Amendments, including Exhibit A-Netscape Public License. Files identified with "Exhibit A-Netscape Public License" are governed by the Netscape Public License Version 1.1.

Additional Terms applicable to the Netscape Public License.

### I. Effect.

These additional terms described in this Netscape Public License -- Amendments shall apply to the Mozilla Communicator client code and to all Covered Code under this License.

II. "Netscape's Branded Code" means Covered Code that Netscape distributes and/or permits others to distribute under one or more trademark(s) which are controlled by Netscape but which are not licensed for use under this License.

III. Netscape and logo.

This License does not grant any rights to use the trademarks "Netscape", the "Netscape N and horizon" logo or the "Netscape lighthouse" logo, "Netcenter", "Gecko", "Java" or "JavaScript", "Smart Browsing" even if such marks are included in the Original Code or Modifications.

IV. Inability to Comply Due to Contractual Obligation.

Prior to licensing the Original Code under this License, Netscape has licensed third party code for use in Netscape's Branded Code. To the extent that Netscape is limited contractually from making such third party code available under this License, Netscape may choose to reintegrate such code into Covered Code without being required to distribute such code in Source Code form, even if such code would otherwise be considered "Modifications" under this License.

V. Use of Modifications and Covered Code by Initial Developer.

V.1. In General.

The obligations of Section 3 apply to Netscape, except to the extent specified in this Amendment, Section V.2 and V.3.

V.2. Other Products.

Netscape may include Covered Code in products other than the Netscape's Branded Code which are released by Netscape during the two (2) years following the release date of the Original Code, without such additional products becoming subject to the terms of this License, and may license such additional products on different terms from those contained in this License.

### V.3. Alternative Licensing.

Netscape may license the Source Code of Netscape's Branded Code, including Modifications incorporated therein, without such Netscape Branded Code becoming subject to the terms of this License, and may license such Netscape Branded Code on different terms from those contained in this License.

### VI. Litigation.

Notwithstanding the limitations of Section 11 above, the provisions regarding litigation in Section 11(a), (b) and (c) of the License shall apply to all disputes relating to this License.

### EXHIBIT A-Netscape Public License.

"The contents of this file are subject to the Netscape Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1998-1999 Netscape Communications Corporation. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the [\_\_\_\_] License), in which case the provisions of [\_\_\_\_\_] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [\_\_\_\_] License and not to allow others to use your version of this file under the NPL, indicate your

decision by deleting the provisions above and replace them with the notice and other provisions required by the [\_\_\_\_] License. If you do not delete the provisions above, a recipient may use your version of this file under either the NPL or the [\_\_\_\_] License."

-----  
MOZILLA  
PUBLIC LICENSE  
Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed



herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. Source Code License.

### 2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer

to use, reproduce,

modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

## 2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the

combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

#### 3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

#### 3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution

Mechanism is maintained by a third party.

### 3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

### 3.4. Intellectual Property Matters

#### (a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

#### (b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

#### (c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed

by this License.

### 3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone,

not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### 3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

## 4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

## 6. Versions of the License.

### 6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

### 6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the

terms applicable to Covered Code created under this License.

### 6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

## 7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

## 9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO



ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

#### 10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

#### 11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

#### 12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is

responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

### 13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as Multiple-Licensed. Multiple-Licensed means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

#### EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is \_\_\_\_\_.

The Initial Developer of the Original Code is \_\_\_\_\_.  
Portions created by \_\_\_\_\_ are Copyright (C) \_\_\_\_\_  
\_\_\_\_\_. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the [\_\_\_\_\_] License), in which case the provisions of [\_\_\_\_\_] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [\_\_\_\_\_] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [\_\_\_\_\_] License. If you do not delete the provisions above, a recipient may

use your

version of this file under either the MPL or the [\_\_\_\_] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### GNU GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the

notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange;  
or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files,  
plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF



MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How

to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

=====  
SDL 2.0 and newer are available under the zlib license :

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====  
=====

#### APACHE HARMONY SUBCOMPONENTS:

Apache Harmony includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

License Notice for ICU4C version 3.4

=====

ICU License - ICU 1.8.1 and later

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2005 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

-----  
All trademarks and registered trademarks mentioned herein are the property of their respective owners.

=====  
=====

License Notice for ICU4J version 4.2.1

=====

ICU4J license - ICU4J 1.8.1 and later

**COPYRIGHT AND PERMISSION NOTICE**

Copyright (c) 1995-2009 International Business  
Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in



the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT

OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

-----

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

=====  
=====

License Notice for FDLIBM version 5.2

=====

Copyright (C) 2004 by Sun Microsystems, Inc. All rights reserved.

Developed

at SunSoft, a Sun Microsystems, Inc. business.

Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

=====

=====  
License Notice for ZLIB version 1.2.3  
=====

(C) 1995-2004 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly      Mark Adler  
jloup@zip.org        madler@alumni.caltech.edu

=====  
Visual C++ Runtime Files  
=====

The file msvc71.dll is the Microsoft(R) C Runtime Library.  
The file msvcp71.dll is the Microsoft Multithreaded Standard C++ Library.

These files are redistributed from Microsoft(R) Visual Studio 7.1 as described in the file "redist.txt" available with that product.

=====  
License

Notice for Bouncy Castle version 1.45  
=====

Copyright (c) 2000 - 2009 The Legion Of The Bouncy Castle (<http://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====  
License Notice for MX4J 3.0.2.  
=====

The MX4J License, Version 1.0

Copyright (c) 2001-2004 by the MX4J contributors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:  
 "This product includes software developed by the MX4J project (<http://mx4j.sourceforge.net>)."  
 Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
  
4. The name "MX4J" must not be used to endorse or promote products derived from this software without prior written permission.  
 For written permission, please contact  
 biorn\_steedom [at] users [dot] sourceforge [dot] net
  
5. Products derived from this software may not be called "MX4J", nor may "MX4J" appear in their name, without prior written permission of Simone Bordet.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MX4J CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This software consists of voluntary contributions made by many individuals on behalf of the MX4J project. For more information on MX4J, please see the MX4J website (<http://mx4j.sourceforge.net/>).

=====

Notice  
for The Independent JPEG Group's JPEG software

---

This distribution may contain software that is based in part on the work of the Independent JPEG Group for which the following notice applies:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.  
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

=====  
=====  
  
Notice for Little CMS  
=====

This distribution may contain software that is based in part on Little cms for which the following notice applies:

Copyright (c) 1998-2007 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====  
=====  
  
Notice for libpng  
=====

This distribution may contain software that is based in part on libpng v1.2.38 for which the following notice applies:

libpng versions 1.2.6, August 15, 2004, through 1.2.38, July 16, 2009, are Copyright (c) 2004, 2006-2009 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5

with the  
following individual added to the list of Contributing Authors:

Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5, October 3, 2002, are  
Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are  
distributed according to the same disclaimer and license as libpng-1.0.6  
with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the  
library or against infringement. There is no warranty that our  
efforts or the library will fulfill any of your particular purposes  
or needs. This library is provided with all faults, and the entire  
risk of satisfactory quality, performance, accuracy, and effort is with  
the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are  
Copyright (c) 1998, 1999, 2000 Glenn Randers-Pehrson, and are  
distributed according to  
the same disclaimer and license as libpng-0.96,  
with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are  
Copyright (c) 1996, 1997 Andreas Dilger  
Distributed according to the same disclaimer and license as libpng-0.88,  
with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are  
Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including,

without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products.

If you use this source code in a product, acknowledgment is not required but would be appreciated.

=====  
=====

Notice for Unicode Character Database

=====

Copyright (c) 1991-2005 Unicode, Inc. All rights reserved. Distributed under the



Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided

that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

=====  
=====

Notice for IETF RFCs prior to 2005  
=====

This distribution may contain software that is based in part on the work of the following IETF RFCs. Portions of the IETF RFC specifications may be included in source code comments for reference, and in accordance with the licensing terms, the licenses are reproduced here:

RFC 1779 - A String Representation of Distinguished Names

RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies

RFC 2251 - Lightweight

Directory Access Protocol (v3)

Copyright (C) The Internet Society (1997). All Rights Reserved.

RFC 2253 - Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names

Copyright (C) The Internet Society (1997). All Rights Reserved.

RFC 2313 - PKCS #1: RSA Encryption

Copyright (C) The Internet Society (1998). All Rights Reserved.

RFC 2315 - PKCS #7: Cryptographic Message Syntax

Copyright (C) The Internet Society (1998). All Rights Reserved.

RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile

Copyright (C) The Internet Society (1999). All Rights Reserved.

RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1

Copyright (C) The Internet Society (1999). All Rights Reserved.

RFC 2781 - UTF-16, an encoding of ISO 10646

Copyright (C) The Internet Society (2000). All Rights Reserved.

RFC 2891 - LDAP Control Extension for Server Side

Sorting of Search Results

Copyright (C) The Internet Society (2000). All Rights Reserved.

RFC 2985 - PKCS #9: Selected Object Classes and Attribute Types

Copyright (C) The Internet Society (2000). All Rights Reserved.

RFC 3161 - Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)

Copyright (C) The Internet Society (2001). All Rights Reserved.

RFC 3280 - Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile

Copyright (C) The Internet Society (2002). All Rights Reserved.

INTERNET-DRAFT - Persistent Search: A Simple LDAP Change Notification Mechanism

<draft-ietf-ldapext-psearch-03.txt>

Copyright (C) The Internet Society (1997-2000). All Rights Reserved.

The following statement applies to each RFC:

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or

assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Notice for IETF RFCs from 2005 onwards

=====

This distribution may contain software that is based in part on the work of the following IETF RFCs. Portions of the IETF RFC specifications may be included in source code comments for reference, and in accordance with the licensing terms, the licenses are reproduced here:

RFC 4122 - A Universally Unique Identifier (UUID) URN Namespace  
Copyright (C) The Internet Society (2005).

The following statement applies to this RFC:

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document

and the information contained herein are provided  
on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE  
REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND  
THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT  
THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR  
ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
PARTICULAR PURPOSE.

=====  
=====

#### Notice for RSA Public-Key Cryptography Standards

=====

Portions of Apache Harmony reference RSA Public-Key Cryptography Standards:  
PKCS#7, PKCS#8 and PKSC#10. Portions of these standards are included  
in Harmony Javadoc for reference, and in accordance with the licensing terms  
for PKCS#7,  
PKCS#8 and PKSC#10, the full copyright statement is here:

Copyright 1991-1993 RSA Laboratories, a division of RSA Data Security, Inc.  
License to copy this document is granted provided that it is identified as  
"RSA Data Security, Inc. Public-Key Cryptography Standards (PKCS)" in all  
material mentioning or referencing this document.

=====  
=====

#### Notice for Apache Yoko

=====

This distribution may contain software that is based in part on Apache Yoko  
for which the following notice applies:

Apache Yoko is an effort undergoing incubation at the Apache Software  
Foundation (ASF). Incubation is required of all newly accepted projects until  
a further review indicates that the infrastructure, communications, and decision  
making process have stabilized in a manner consistent with other successful  
ASF projects. While incubation

status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

<http://incubator.apache.org/yoko/>

=====  
=====

Notice for the Eclipse JDT Core Batch Compiler

=====

The license terms for the Eclipse JDT Core Batch Compiler are available at  
<http://www.eclipse.org/legal/epl-v10.html>

Source code for the Eclipse JDT Core Batch Compiler can be found at:  
<http://download.eclipse.org/eclipse/downloads/>

Navigate to the version of ECJ used in this distribution, and the source is provided in a ZIP file in the section titled "JDT Core Batch Compiler".

=====  
=====

License

notice for the DejaVu fonts package

=====

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.  
Glyphs imported from Arev fonts are (c) Tavmjung Bah (see below)

Bitstream Vera Fonts Copyright

-----

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated

documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright  
-----

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and

associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including  
without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE  
FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free . fr.

=====  
=====

License  
notice for ASM

=====

ASM: a very small and fast Java bytecode manipulation framework  
Copyright (c) 2000-2005 INRIA, France Telecom  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its  
contributors may be used to endorse or promote products derived from  
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
AND ANY  
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE  
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF  
THE POSSIBILITY OF SUCH DAMAGE.

=====

License notice for Junit

=====

This distribution may contain a jar file based on Junit 4.2.1 for  
which the following notice applies:



Copyright

2001-2009 Kent Beck <kent@threeriversinstitute.org>

Copyright 2000-2009 Erich Gamma <erich\_gamma@acm.org>

Copyright 2006-2009 David Saff <david@saff.net>

Common Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

## 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:

- i) changes to the Program, and

- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not

include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to

reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

#### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor

("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may

participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such

Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

=====  
=====

The

Apache Harmony DRLVM uses the Apache Portable Runtime which includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

From strings/apr\_fnmatch.c, include/apr\_fnmatch.h, misc/unix/getopt.c, file\_io/unix/mktemp.c, strings/apr\_strings.c:

/\*

\* Copyright (c) 1987, 1993, 1994

\* The Regents of the University of California. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\*

notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* This product includes software developed by the University of

\* California, Berkeley and its contributors.

\* 4. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING,

BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

From network\_io/unix/inet\_ntop.c, network\_io/unix/inet\_pton.c:

/\* Copyright (c) 1996 by Internet Software Consortium.

\*

\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS  
\* ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES  
\* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE  
\* CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT,  
\* OR CONSEQUENTIAL  
\* DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR  
\* PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS  
\* ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS  
\* SOFTWARE.

From dso/aix/dso.c:

\* Based on libdl (dlfcn.c/dlfcn.h) which is  
\* Copyright (c) 1992,1993,1995,1996,1997,1988  
\* Jens-Uwe Mager, Helios Software GmbH, Hannover, Germany.  
\*  
\* Not derived from licensed software.  
\*  
\* Permission is granted to freely use, copy, modify, and redistribute  
\* this software, provided that the author is not construed to be liable  
\* for any results of using the software, alterations are clearly marked  
\* as such, and this notice is not modified.

From strings/apr\_strnatcmp.c, include/apr\_strings.h:

strnatcmp.c -- Perform 'natural order' comparisons of strings in C.  
Copyright (C) 2000 by Martin Pool <mbp@humblebug.org.au>

This software is provided 'as-is', without any express or implied  
warranty.

In no event will the authors be held liable for any damages  
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,  
including commercial applications, and to alter it and redistribute it  
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not  
claim that you wrote the original software. If you use this software  
in a product, an acknowledgment in the product documentation would be  
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be  
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

From strings/apr\_sprintf.c:

\*  
\* cvt - IEEE floating point formatting routines.  
\* Derived from UNIX V7, Copyright(C) Caldera International Inc.  
\*

Copyright(C) Caldera International Inc. 2001-2002. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code and documentation must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed or owned by Caldera International, Inc.

Neither the name of Caldera International, Inc. nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

USE OF THE SOFTWARE PROVIDED FOR UNDER THIS LICENSE BY CALDERA INTERNATIONAL, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CALDERA INTERNATIONAL, INC. BE LIABLE FOR ANY DIRECT, INDIRECT INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====



=====  
License notice for Hamcrest  
=====

BSD License

Copyright (c) 2000-2006,  
www.hamcrest.org  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of  
conditions and the following disclaimer. Redistributions in binary form must reproduce  
the above copyright notice, this list of conditions and the following disclaimer in  
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse  
or promote products derived from this software without specific prior written  
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY  
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO  
EVENT  
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
LIMITED  
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR  
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN  
ANY  
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH  
DAMAGE.

=====  
=====  
=====  
libc++ License

---

The libc++ library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

---

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2017 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS

FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====  
Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Copyright (C) 2012-2016 Free Software Foundation, Inc.

# Redistribution and use in source and binary forms, with or without  
# modification, are permitted provided that the following conditions are  
# met:

# (1) Redistributions of source code must retain the above copyright  
# notice, this list of conditions and the following disclaimer.

# (2) Redistributions in binary form must reproduce the above copyright  
# notice, this list of conditions and the following disclaimer in  
# the documentation and/or other materials provided with the  
# distribution.

# (3) The name of the author may not be used to  
# endorse or promote products derived from this software without  
# specific prior written permission.

# THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR

# IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
# DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE  
# FOR ANY DIRECT,  
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING  
# IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
# POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2005-2010 Sam Stephenson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2019 Microsoft Corporation, Daan Leijen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2009-2015 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

## 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor"

means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

## 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively

excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

#### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

#### 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED

INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,

OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such

Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement

Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.



This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

```
=====
== NOTICE file corresponding to the section 4 d of ==
== the Apache License, Version 2.0, ==
== in this case for the Kotlin Compiler distribution. ==
=====
```

Kotlin Compiler

Copyright 2010-2020 JetBrains s.r.o and respective authors and developers

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This license applies to all parts of Dart that are not externally maintained libraries. The external maintained libraries used by Dart are:

7-Zip - in third\_party/7zip

JSCORE - in runtime/third\_party/jscre

Ant - in third\_party/apache\_ant

args4j - in third\_party/args4j

bzip2 - in third\_party/bzip2  
dromaeo - in samples/third\_party/dromaeo  
Eclipse - in third\_party/eclipse  
gsutil = in third\_party/gsutil  
Guava - in third\_party/guava  
hamcrest - in third\_party/hamcrest  
HttpLib2 - in samples/third\_party/httpLib2  
JSON - in third\_party/json  
JUnit - in third\_party/junit  
Oauth - in samples/third\_party/oauth2client  
Rhino - in third\_party/rhino  
weberknecht - in third\_party/weberknecht

The libraries may have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Copyright 2012, the Dart project authors. All rights reserved.  
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

<https://www.eclipse.org/jgit/>

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

libffi - Copyright (c) 1996-2014 Anthony Green, Red Hat, Inc and others.

See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

Copyright (C) 2009 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation,



modification and use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor architecture, in executable form or suitable for input to an assembler, loader, linker and/or execution phase. Notwithstanding that, Target Code does not include data in any format that is used as a compiler intermediate representation, or used for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in non-intermediate languages designed for human-written code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use of source code generators and preprocessors need not be considered part of the Compilation Process, since the Compilation Process can be understood as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-compatible software, or if it is done without using any work based on GCC. For example, using non-GPL-compatible Software to optimize any GCC intermediate representations would not qualify as an Eligible Compilation Process.

#### 1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime Library with Independent Modules, even if such propagation would otherwise violate the terms of GPLv3, provided that all Target Code was generated by Eligible Compilation Processes. You may then convey such a combination under terms of your choice, consistent with the licensing of the Independent Modules.

#### 2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements of the license of GCC.

Hopefully that text is self-explanatory. If it isn't, you need to speak to your lawyer, or the Free Software Foundation.  
JSON

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Eclipse Public License, Version 1.0 (EPL-1.0)

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

## 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and

documentation distributed under this Agreement, and

- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and  
(ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.  
No hardware per se is licensed hereunder.

- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
- i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
  - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and



b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

#### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses,

damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

- a) promptly notify the Commercial Contributor in writing of such claim,
- and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor.

If that Commercial Contributor

then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

#### 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR

IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may

only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated

in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The MIT License

Copyright (C) 2011-2013 Vojta Jna and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
<http://www.slf4j.org>

MIT License

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.96 wire-profiles 3.7.1

### 1.96.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2016 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1569088931_1676645219.9753337/0/wire-profiles-3-7-1-sources-
jar/com/squareup/wire/java/AdapterConstant.kt
* /opt/cola/permits/1569088931_1676645219.9753337/0/wire-profiles-3-7-1-sources-
```

```
jar/com/squareup/wire/java/internal/ProfileParser.kt
* /opt/cola/permits/1569088931_1676645219.9753337/0/wire-profiles-3-7-1-sources-
jar/com/squareup/wire/java/Profile.kt
*
/opt/cola/permits/1569088931_1676645219.9753337/0/wire-profiles-3-7-1-sources-
jar/com/squareup/wire/java/internal/ProfileFileElement.kt
* /opt/cola/permits/1569088931_1676645219.9753337/0/wire-profiles-3-7-1-sources-
jar/com/squareup/wire/java/internal/TypeConfigElement.kt
```

## 1.97 wire-grpc-server-generator 3.7.1

### 1.97.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/KotlinGrpcGenerator.kt
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/BlockingStubGenerator.kt
*
/opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/ServiceDescriptorGenerator.kt
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/StubGenerator.kt
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/ImplBaseGenerator.kt
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/LegacyAdapterGenerator.kt
* /opt/cola/permits/1569088938_1676645221.8978014/0/wire-grpc-server-generator-3-7-1-sources-
jar/com/squareup/wire/kotlin/grpcserver/MethodDescriptorGenerator.kt
```

# 1.98 wire-swift-generator 3.7.1

## 1.98.1 Available under license :

Apache-2.0

# 1.99 wire-java-generator 3.7.1

## 1.99.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1569088940_1676645225.320955/0/wire-java-generator-3-7-1-sources-1-
jar/com/squareup/wire/java/JavaGenerator.java
```

# 1.100 amazon-kinesis-client 2.4.5

## 1.100.1 Available under license :

AmazonKinesisClientLibrary

Copyright 2012-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and



attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.101 netty-nio-client 2.19.2

## 1.101.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

# 1.102 commons-fileupload 1.5

## 1.102.1 Available under license :

Apache Commons FileUpload  
Copyright 2002-2023 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership



of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.103 tre 1.2.3-r4

## 1.103.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

-----  
Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----

Authors/contributors include:

A. Wilcox  
Ada Worcester  
Alex Dowad  
Alex Suykov  
Alexander Monakov  
Andre McCurdy  
Andrew Kelley  
Anthony G. Basile  
Aric Belsito  
Arvid Picciani  
Bartosz Brachaczek  
Benjamin Peterson  
Bobby Bingham  
Boris Brezillon  
Brent Cook  
Chris Spiegel

Clment Vasseur  
Daniel Micay  
Daniel Sabogal  
Daurnimator  
David Carlier  
David Edelsohn  
Denys Vlasenko  
Dmitry Ivanov  
Dmitry V. Levin  
Drew DeVault  
Emil Renner Berthing  
Fangrui Song  
Felix Fietkau  
Felix Janda  
Gianluca Anzolin  
Hauke Mehtens  
He X  
Hiltjo Posthuma  
Isaac Dunham  
Jaydeep Patil  
Jens Gustedt  
Jeremy Huntwork  
Jo-Philipp Wich  
Joakim Sindholt  
John Spencer  
Julien Ramseier  
Justin Cormack  
Kaarle Ritvanen  
Khem Raj  
Kylie McClain  
Leah Neukirchen  
Luca Barbato  
Luka  
Perkov  
M Farkas-Dyck (Strake)  
Mahesh Bodapati  
Markus Wichmann  
Masanori Ogino  
Michael Clark  
Michael Forney  
Mikhail Kremnyov  
Natanael Copa  
Nicholas J. Kain  
orc  
Pascal Cuoq  
Patrick Oppenlander  
Petr Hosek  
Petr Skocik

Pierre Carrier  
Reini Urban  
Rich Felker  
Richard Pennington  
Ryan Fairfax  
Samuel Holland  
Segev Finer  
Shiz  
sin  
Solar Designer  
Stefan Kristiansson  
Stefan O'Rear  
Szabolcs Nagy  
Timo Ters  
Trutz Behn  
Valentin Ochs  
Will Dietz  
William Haddon  
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (`src/string/aarch64/*`) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (`src/crypt/crypt_des.c`) is

Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/crypt/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsort.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86\_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the elcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (`include/*` and `arch/*/bits/*`) and crt files intended to be linked into applications (`crt/*`, `ldso/dlstart.c`, and `arch/*/crt_arch.h`) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham  
John Spencer  
Nicholas J. Kain  
Rich Felker  
Richard Pennington  
Stefan Kristiansson



Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated

the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

## 1.104 pgv-java-stub 0.6.1

### 1.104.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.105 apache-log4j 2.20.0

## 1.105.1 Available under license :

Apache Log4j Core

Copyright 1999-2012 Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java

Copyright 2005-2006 Tim Fennell

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.



6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.106 log4j-api 2.20.0

### 1.106.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Log4j 1.x Compatibility API  
Copyright 1999-2023 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

# 1.107 log4j-slf4j2-impl 2.20.0

## 1.107.1 Available under license :

Apache Log4j SLF4J 2.0 Binding  
Copyright 1999-2023 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You



institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.108 error\_prone\_annotations 2.18.0

## 1.108.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2015 The Error Prone Authors.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

\* /opt/cola/permits/1606613656\_1679417956.422988/0/error-prone-annotations-2-18-0-1-jar/META-INF/maven/com.google.errorprone/error\_prone\_annotations/pom.xml

# 1.109 opentelemetry-proto 0.11.0

## 1.109.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.110 json-java 20201115

## 1.110.1 Available under license :

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.



THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.111 everit-json-schema 1.12.2

### 1.111.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright (C) 2011 Everit Kft. (<http://www.everit.org>)

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

-->

Found in path(s):

\* /opt/cola/permits/1611898203\_1680235707.0150697/0/everit-json-schema-1-12-2-sources-4-jar/META-INF/maven/com.github.erosb/everit-json-schema/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1611898203\_1680235707.0150697/0/everit-json-schema-1-12-2-sources-4-jar/org/everit/json/schema/JSONPointerException.java

\* /opt/cola/permits/1611898203\_1680235707.0150697/0/everit-json-schema-1-12-2-sources-4-jar/org/everit/json/schema/JSONPointer.java

No license file was found, but licenses were detected in source scan.

/\*

Copyright (c) 2006 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1611898203\_1680235707.0150697/0/everit-json-schema-1-12-2-sources-4-jar/org/everit/json/schema/internal/JSONWriter.java

# 1.112 classgraph 4.8.120

## 1.112.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>The MIT License (MIT)</name>

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/META-INF/maven/io.github.classgraph/classgraph/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: R. Kempees

\*

\* With contributions from @cpierceworld (#414)

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2017 R. Kempees (contributed to the ClassGraph project)

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

\* documentation files (the "Software"), to deal in the Software without restriction, including without

\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following

\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial

\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY

OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/WebsphereLibertyClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: @jacobg on GitHub

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 @jacobg, Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES

OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/AntClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.  
 \*  
 \* Author: Luke Hutchison  
 \*  
 \* Hosted at: <https://github.com/classgraph/classgraph>  
 \*  
 \* --  
 \*  
 \* The MIT License (MIT)  
 \*  
 \* Copyright (c) 2020 Luke Hutchison  
 \*  
 \* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
 \* documentation files (the "Software"), to deal in the Software without restriction, including without  
 \* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
 \* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
 \* conditions:  
 \*  
 \* The above copyright notice and this permission notice shall be included in all copies or substantial  
 \* portions of the Software.  
 \*  
 \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
 \* IMPLIED, INCLUDING BUT NOT  
 \* LIMITED TO THE WARRANTIES OF  
 \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
 \* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
 \* OR OTHER LIABILITY, WHETHER IN  
 \* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
 \* WITH THE SOFTWARE OR THE USE  
 \* OR OTHER DEALINGS IN THE SOFTWARE.  
 \*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessReader.java  
 \* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/ArraySlice.java  
 \* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/PathSlice.java  
 \* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/FileSlice.java  
 \* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/reader/ClassfileReader.java  
 \*  
 /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessByteBufferReader.java  
 \* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
 jar/nonapi/io/github/classgraph/fileslice/reader/SequentialReader.java

```
* /opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessArrayReader.java
* /opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessFileChannelReader.java
* /opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/fileslice/Slice.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* This file is part of ClassGraph.
```

```
*
```

```
* Author: Johno Crawford (johno@sulake.com)
```

```
*
```

```
* Hosted at: https://github.com/classgraph/classgraph
```

```
*
```

```
* --
```

```
*
```

```
* The MIT License (MIT)
```

```
*
```

```
* Copyright (c) 2016 Johno Crawford
```

```
*
```

```
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
```

```
* documentation files (the "Software"), to deal in the Software without restriction, including without
```

```
* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
```

```
* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
```

```
* conditions:
```

```
*
```

```
* The above copyright notice and this permission notice shall be included in all copies or substantial
```

```
* portions of the Software.
```

```
*
```

```
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT
```

```
* LIMITED TO
```

```
THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO
```

```
* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN
```

```
* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE
```

```
* OR OTHER DEALINGS IN THE SOFTWARE.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/concurrency/AutoCloseableExecutorService.java
```

```
* /opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/concurrency/SimpleThreadFactory.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * This file is part of ClassGraph.
 *
 * Author: Sergey Beshpalov
 *
 * Hosted at: https://github.com/classgraph/classgraph
 *
 * --
 *
 * The MIT License (MIT)
 *
 * Copyright (c) 2017 Sergey Beshpalov
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
 * documentation files (the "Software"), to deal in the Software without restriction, including without
 * limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
 * the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
 * conditions:
 *
 * The above copyright notice and this permission notice shall be included in all copies or substantial
 * portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT
 * LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
 * EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
 * OR OTHER LIABILITY, WHETHER IN
 * AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE
 * OR OTHER DEALINGS IN THE SOFTWARE.
 */
```

Found in path(s):

```
*/opt/cola/permits/1611898211_1680234953.7192042/0/classgraph-4-8-120-sources-2-
jar/nonapi/io/github/classgraph/classloaderhandler/WebsphereTraditionalClassLoaderHandler.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * This file is part of ClassGraph.
 *
 * Author: Luke Hutchison
 *
 * Hosted at: https://github.com/classgraph/classgraph
 *
 * --
```

\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2019 Luke Hutchison  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/types/Parser.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/TypeVariableSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/MethodParameterInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/ParameterizedTypeImpl.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/EquinoxContextFinderClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/FallbackClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/WeblogicClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/types/TypeUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassInfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-



jar/io/github/classgraph/ModuleInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classpath/SystemJarFinder.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/JSONSerializer.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/FieldInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/TomcatWebappClassLoaderBaseHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/fastzipfilereader/ZipFileSlice.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/ClassFieldCache.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/TypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassGraphClassLoader.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassRefTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/ReflectionUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/HierarchicalTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassGraphException.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/ParentLastDelegationOrderTestClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClasspathElementFileDir.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/OSGiDefaultClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/URLPathEncoder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/JSONDeserializer.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classpath/ClassLoaderFinder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classpath/ClassLoaderOrder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/AnnotationClassRef.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/HasName.java

\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ModuleInfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ClasspathElement.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/ClassLoaderHandlerRegistry.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/Utils/JarUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/JSONParser.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/PlexusClassWorldsClassRealmClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/Resource.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ClasspathElementPathDir.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/TypeResolutions.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/BaseTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/URLClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/MethodTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/PotentiallyUnmodifiableList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/concurrency/WorkQueue.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/JSONReference.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ClassRefOrTypeVariableSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ClasspathElementZip.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/Utils/FastPathResolver.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/recycler/Resettable.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/FieldInfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/concurrency/InterruptedException.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/fastzipfilereader/LogicalZipFile.java  
\*

/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/InfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ReferenceTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/ReferenceEqualityKey.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/ClassFields.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classpath/ClasspathFinder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/concurrency/SingletonMap.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/AnnotationParameterValueList.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/ClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/fastzipfilereader/FastZipEntry.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/EquinoxClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ModuleRef.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/ClassGraphClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/PackageInfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ClasspathElementModule.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/TypeArgument.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/scanspec/ScanSpec.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ArrayTypeSignature.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classpath/ClasspathOrder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classpath/CallStackReader.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/AnnotationEnumValue.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/fastzipfilereader/PhysicalZipFile.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/json/Id.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-

jar/io/github/classgraph/MethodInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/TypeParameter.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classpath/ModuleFinder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/MappableInfoList.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/FileUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/AnnotationParameterValue.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/Classfile.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/JSONObject.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/LogNode.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/VersionFinder.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ClassGraph.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/GraphvizDotfileGenerator.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/JSONArray.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/Utils/StringUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ArrayClassInfo.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/types/ParseException.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/UnoOneJarClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/FieldTypeInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/JPMSCClassLoaderHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/fastzipfilereader/NestedJarHandler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/json/JSONUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/ScanResultObject.java  
\*  
/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/io/github/classgraph/AnnotationInfoList.java

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ModulePathInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/Utils/CollectionUtils.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ScanResult.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ObjectTypedValueWrapper.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/PackageInfo.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/AnnotationInfo.java  
\*

/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/recycler/Recycler.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/scanspec/AcceptReject.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/recycler/RecycleOnClose.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/Scanner.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ModuleReaderProxy.java  
\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/MethodInfoList.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2021 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\*/opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/CxfContainerClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison (luke.hutch@gmail.com)

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT

\* LIMITED

TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/io/github/classgraph/ResourceList.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Michael J. Simons

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

\* documentation files (the "Software"), to deal in the Software without restriction, including without

\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following

\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial

\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES

\* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/SpringBootRestartClassLoaderHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*  
\* Author: Harith Elrifaie  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2017 Harith Elrifaie  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF  
\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
\* OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/FelixClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* This file is part of ClassGraph.  
\*  
\* Author: Luke Hutchison  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2019 Luke Hutchison, with significant contributions from Davy De Duerpel



\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED,  
INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-  
jar/nonapi/io/github/classgraph/classloaderhandler/JBossClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* This file is part of ClassGraph.  
\*  
\* Author: @mcollovati  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2019 @mcollovati, contributed to the ClassGraph project  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.

\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1611898211\_1680234953.7192042/0/classgraph-4-8-120-sources-2-jar/nonapi/io/github/classgraph/classloaderhandler/QuarkusClassLoaderHandler.java

## 1.113 jimfs 1.1

### 1.113.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.114 protobuf-java 3.22.2

### 1.114.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
```

```
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MapEntry.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/TextFormatParseLocation.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/Internal.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/BufferAllocator.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ProtoSyntax.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/FieldInfo.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ByteOutput.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/IterableByteBufferInputStream.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MutabilityOracle.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/Parser.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/RpcController.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/GeneratedMessageInfoFactory.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/BinaryWriter.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/SchemaUtil.java
*
```

/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MapFieldLite.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Service.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/GeneratedMessageLite.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionLite.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionRegistryLite.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/DynamicMessage.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Protobuf.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MessageInfo.java  
\*  
/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/NewInstanceSchema.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/InvalidProtocolBufferException.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionSchemaFull.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/TextFormat.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ManifestSchemaFactory.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/LongArrayList.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/AbstractParser.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/NewInstanceSchemaLite.java  
\*  
/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionSchemas.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/PrimitiveNonBoxingCollection.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MapEntryLite.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/NewInstanceSchemaFull.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MessageSchema.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/DescriptorMessageInfoFactory.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Writer.java



\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MapFieldSchemaLite.java  
\*  
/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/CompileTimeConstant.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ByteBufferWriter.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Utf8.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/BlockingRpcChannel.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MapFieldSchema.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionSchema.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionRegistryFactory.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/LazyStringArrayList.java  
\*  
/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/WireFormat.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/UnknownFieldSet.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/UnknownFieldSchema.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/NioByteString.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/FieldType.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/GeneratedMessage.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/TextFormatParseInfoTree.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/RawMessageInfo.java  
\*  
/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/RpcChannel.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MapFieldSchemas.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/RopeByteString.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/JavaType.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/CodedOutputStreamWriter.java  
\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-

```

jar/com/google/protobuf/ProtobufArrayList.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/CanIgnoreReturnValue.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageLiteOrBuilder.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/LazyStringList.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/FieldSet.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ByteString.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/LazyFieldLite.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/Extension.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/DoubleArrayList.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ListFieldSchema.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/TypeRegistry.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageOrBuilder.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MapFieldSchemaFull.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageInfoFactory.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UnsafeUtil.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MapField.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/RpcUtil.java

```

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/StructuralMessageInfo.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/RepeatedFieldBuilder.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/BooleanArrayList.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/IntArrayList.java

\*

/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ProtocolStringList.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/UnsafeByteOperations.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/SingleFieldBuilderV3.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/NewInstanceSchemas.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExtensionRegistry.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Android.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Reader.java

\*

/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/AllocatedBuffer.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/DiscardUnknownFieldsParser.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/AbstractMessageLite.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/ExperimentalApi.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/MessageSetSchema.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/InlineMe.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Message.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/OneofInfo.java

\*

/opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/Descriptors.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/TextFormatEscaper.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-jar/com/google/protobuf/CodedOutputStream.java

\* /opt/cola/permits/1614739947\_1680235541.8595586/0/protobuf-java-3-22-2-sources-

```

jar/com/google/protobuf/BinaryReader.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UnknownFieldSetLiteSchema.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/FloatArrayList.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/SchemaFactory.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/BlockingService.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ExtensionSchemaLite.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageLite.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/ArrayDecoders.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/Schema.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/Java8Compatibility.java
*
/opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/GeneratedMessageV3.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/CheckReturnValue.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/UninitializedMessageException.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/1614739947_1680235541.8595586/0/protobuf-java-3-22-2-sources-
jar/com/google/protobuf/AbstractProtobufList.java

```

## 1.115 pgv-java-stub 0.9.1

### 1.115.1 Available under license :

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate



comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.116 annotations 2.19.2

## 1.116.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[ ]"  
replaced with your own identifying information. (Don't include  
the brackets!) The text should be enclosed in the appropriate  
comment syntax for the file format. We also recommend that a  
file or class name and description of purpose be included on the  
same "printed page" as the copyright notice for easier  
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.117 protocol-core 2.19.2

### 1.117.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work



or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.118 netty-transport-classes-epoll

## 4.1.90.Final

### 1.118.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * Set the {@code TCP_MD5SIG} option on the socket. See {@code linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java
```

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2021 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
```

~  
~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.  
-->

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-transport-classes-epoll/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2021 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollDomainDatagramChannelConfig.java

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollDomainDatagramChannel.java

\*

/opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/SegmentedDatagramPacket.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/NativeStaticallyReferencedJniMethods.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/LinuxSocket.java

No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,  
# version 2.0 (the "License"); you may not use this file except in compliance  
# with the License. You may obtain a copy of the License at:  
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/META-INF/native-image/io.netty/netty-transport-classes-epoll/native-image.properties

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/Native.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/AbstractEpollChannel.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollChannelOption.java  
\*  
/opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollTcpInfo.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/NativeDatagramPacketArray.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollEventLoop.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollDatagramChannel.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollSocketChannel.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannel.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/package-info.java  
\*  
/opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/Epoll.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollEventLoopGroup.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>

```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * Set the { @code TCP_MD5SIG } option on the socket. See { @code linux/tcp.h } for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
*/
*/
/**
 * Set the { @code TCP_QUICKACK } option on the socket.
 * See TCP_QUICKACK
 * for more details.
*/

```

Found in path(s):

```

* /opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/EpollSocketChannelConfig.java

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
*/

```

Found in path(s):

```

* /opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/EpollChannelConfig.java
* /opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/AbstractEpollStreamChannel.java
*

```

```

/opt/cola/permits/1620675266_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-
jar/io/netty/channel/epoll/EpollServerChannelConfig.java

```



\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollRecvByteAllocatorStreamingHandle.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/TcpMd5Util.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollEventArray.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollMode.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/AbstractEpollServerChannel.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollServerDomainSocketChannel.java  
\*  
/opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollRecvByteAllocatorHandle.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollDomainSocketChannelConfig.java  
\* /opt/cola/permits/1620675266\_1680116307.60289/0/netty-transport-classes-epoll-4-1-90-final-sources-jar/io/netty/channel/epoll/EpollDomainSocketChannel.java

# 1.119 third-party-jackson-core 2.19.2

## 1.119.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](https://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.  
It was originally written by Tatu Saloranta ([tatu.saloranta@iki.fi](mailto:tatu.saloranta@iki.fi)), and has  
been in development since 2007.  
It is currently developed by a community of developers.

#### ## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0  
To find the details that apply to this artifact see the accompanying LICENSE file.

#### ## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included  
in some artifacts (usually source distributions); but is always available  
from the source code management (SCM) system project uses.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and



attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.120 json-utils 2.19.2

## 1.120.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

# 1.121 cloudwatch 2.19.2

## 1.121.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation



This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[ ]"  
replaced with your own identifying information. (Don't include  
the brackets!) The text should be enclosed in the appropriate  
comment syntax for the file format. We also recommend that a  
file or class name and description of purpose be included on the  
same "printed page" as the copyright notice for easier  
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.122 client-spi 2.19.2

### 1.122.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.



5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.123 utils 2.19.2

## 1.123.1 Available under license :

MIT License

Copyright (c) 2019 TypeScript ESLint and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.124 aws-cbor-protocol 2.19.2

### 1.124.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.125 sdk-core 2.19.2

### 1.125.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).



## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.126 netty-codec-socks 4.1.90.Final

### 1.126.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

`*/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksCmdType.java`

```
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5AddressType.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/SocksVersion.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksAuthStatus.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandType.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthStatus.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5AuthMethod.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksRequestType.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksResponseType.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksProtocolVersion.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandStatus.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksSubnegotiationVersion.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksAddressType.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksMessageType.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksAuthScheme.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksCmdStatus.java
No license file was found, but licenses were detected in source scan.
```

<!--

~ Copyright 2012 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,  
~ version 2.0 (the "License"); you may not use this file except in compliance  
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.

-->

Found in path(s):

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-codec-socks/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/package-info.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/package-info.java

\*

/opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequestDecoder.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/AbstractSocks5Message.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerEncoder.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/AbstractSocks4Message.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponseDecoder.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponseDecoder.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponseDecoder.java

\*

/opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/AbstractSocksMessage.java

\* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5ClientEncoder.java

```
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5ServerEncoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v4/Socks4ClientEncoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequestDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v4/Socks4Message.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequestDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5Message.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5AddressDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5AddressEncoder.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/SocksPortUnificationServerHandler.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*/
```



- \* <https://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
- \* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
- \* License for the specific language governing permissions and limitations
- \* under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandStatus.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandResponse.java
- \*
- /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponse.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandResponse.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerDecoder.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksMessage.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksAuthResponse.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksCmdRequest.java
- \*
- /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksInitRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/UnknownSocksRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandResponse.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/package-info.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socks/SocksMessageEncoder.java
- \*
- /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthRequest.java
- \* /opt/cola/permits/1620675433\_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-

```

jar/io/netty/handler/codec/socks/SocksAuthRequestDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandRequest.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequest.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksInitRequestDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v4/Socks4CommandType.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksAuthRequest.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksAuthResponseDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksInitResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/UnknownSocksResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksCmdResponseDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksCmdResponse.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/package-info.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialRequest.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/SocksMessage.java
*
/opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksCommonUtils.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v4/Socks4ClientDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequest.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksInitResponseDecoder.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksRequest.java

```

```
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialResponse.java
* /opt/cola/permits/1620675433_1680116262.0139341/0/netty-codec-socks-4-1-90-final-sources-
jar/io/netty/handler/codec/socks/SocksCmdRequestDecoder.java
```

# 1.127 netty-resolver 4.1.90.Final

## 1.127.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2021 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/HostsFileEntriesProvider.java
No license file was found, but licenses were detected in source scan.
```

```
<!--
~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-resolver/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/AddressResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/InetSocketAddressResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/DefaultAddressResolverGroup.java

\*

/opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/InetAddressResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/DefaultHostsFileEntriesResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/AbstractAddressResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/HostsFileEntriesResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/DefaultNameResolver.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/HostsFileParser.java

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-jar/io/netty/resolver/CompositeNameResolver.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2017 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-  
jar/io/netty/resolver/HostsFileEntries.java  
\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-  
jar/io/netty/resolver/ResolvedAddressTypes.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675276\_1680234965.994694/0/netty-resolver-4-1-90-final-sources-  
jar/io/netty/resolver/RoundRobinInetAddressResolver.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\*/

```
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/SimpleNameResolver.java
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/NoopAddressResolverGroup.java
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/NoopAddressResolver.java
*
/opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/package-info.java
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/NameResolver.java
* /opt/cola/permits/1620675276_1680234965.994694/0/netty-resolver-4-1-90-final-sources-
jar/io/netty/resolver/AddressResolverGroup.java
```

# 1.128 netty-handler-proxy 4.1.90.Final

## 1.128.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1620675501_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/META-
```

INF/maven/io.netty/netty-handler-proxy/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/ProxyHandler.java

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/ProxyConnectionEvent.java

\*

/opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/HttpProxyHandler.java

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/package-info.java

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/Socks4ProxyHandler.java

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/ProxyConnectException.java

\* /opt/cola/permits/1620675501\_1680234968.1694567/0/netty-handler-proxy-4-1-90-final-sources-jar/io/netty/handler/proxy/Socks5ProxyHandler.java

## 1.129 netty-handler 4.1.90.Final

### 1.129.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2022 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version

\* 2.0 (the "License"); you may not use this file except in compliance with the

\* License. You may obtain a copy of the License at:

\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslCertificateCompressionConfig.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2017 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the



\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/Conscrypt.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SniCompletionEvent.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslCloseCompletionEvent.java  
\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OptionalSslHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ocsp/OcspClientHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslCompletionEvent.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ocsp/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/AbstractSniHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JdkAlpnSslEngine.java  
\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JdkAlpnSslUtils.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslClientHelloHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/flush/package-info.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/PemEncoded.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslServerContext.java
- \*
- /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/PemPrivateKey.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/PemValue.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslEngine.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/Java8SslUtils.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/flush/FlushConsolidationHandler.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslCertificateException.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/DelegatingSslContext.java
- \*
- /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/PemX509Certificate.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslClientContext.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslContext.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/X509TrustManagerWrapper.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/flow/package-info.java
- \* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslSessionContext.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslProvider.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/CipherSuiteConverter.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkSslEngine.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/RuleBasedIpFilter.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/UniqueIpFilter.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/IpFilterRule.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/IpSubnetFilterRule.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/InsecureTrustManagerFactory.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/Java7SslParametersUtils.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkAlpnApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SupportedCipherSuiteFilter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/CipherSuiteFilter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/IpFilterRuleType.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkSslServerContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslEngine.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslUtils.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ipfilter/AbstractRemoteAddressFilter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/SimpleTrustManagerFactory.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkSslContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/ApplicationProtocolConfig.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/ApplicationProtocolUtil.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkBaseApplicationProtocolNegotiator.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslSessionStats.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/IdentityCipherSuiteFilter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/JdkApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/PemReader.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslEngineMap.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/LazyX509Certificate.java  
\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslDefaultApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslServerContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslClientContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SniHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficCounter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslServerSessionContext.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JettyAlpnSslEngine.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/BouncyCastleSelfSignedCertGenerator.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslNpnApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSsl.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JdkNpnApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/OpenJdkSelfSignedCertGenerator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslApplicationProtocolNegotiator.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JdkDefaultApplicationProtocolNegotiator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JettyNpnSslEngine.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/FingerprintTrustManagerFactory.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/ThreadLocalInsecureRandom.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ipfilter/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/JdkSslClientContext.java  
No license file was found, but licenses were detected in source scan.  
  
/\*  
\* Copyright 2011 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/AbstractTrafficShapingHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslContextBuilder.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ClientAuth.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/util/LazyJavaxX509Certificate.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNames.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiationHandler.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslSessionTicketKey.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

// Try the OpenJDK's proprietary implementation.

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/util/SelfSignedCertificate.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version

\* 2.0 (the "License"); you may not use this file except in compliance with the

\* License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/flow/FlowControlHandler.java

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2012 The Netty Project

~  
~ The Netty Project licenses this file to you under the Apache License,  
~ version 2.0 (the "License"); you may not use this file except in compliance  
~ with the License. You may obtain a copy of the License at:  
~  
~ <https://www.apache.org/licenses/LICENSE-2.0>  
~  
~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.  
-->

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-handler/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/timeout/WriteTimeoutException.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/timeout/ReadTimeoutException.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/timeout/WriteTimeoutHandler.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/stream/ChunkedFile.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/logging/package-info.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-



jar/io/netty/handler/stream/ChunkedWriteHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/TimeoutException.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/stream/ChunkedInput.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/logging/LoggingHandler.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/stream/ChunkedStream.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/stream/ChunkedNioFile.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/IdleStateHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/ReadTimeoutHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/package-info.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/IdleStateEvent.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/stream/ChunkedNioStream.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/NotSslRecordException.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/logging/LogLevel.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/stream/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/timeout/IdleState.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2022 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslCertificateCompressionAlgorithm.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/BouncyCastlePemReader.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2019 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/SslMasterKeyHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/OpenSslPrivateKeyMethod.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/KeyManagerFactoryWrapper.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/TrustManagerFactoryWrapper.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/PseudoRandomFunction.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/X509KeyManagerWrapper.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/address/package-info.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-  
jar/io/netty/handler/ssl/util/SimpleKeyManagerFactory.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/address/DynamicAddressConnectHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/package-info.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2023 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/StacklessSSLHandshakeException.java

No license file was found, but licenses were detected in source scan.

```
The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/META-INF/native-image/io.netty/netty-handler/native-image.properties

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2023 The Netty Project

\*

```
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
```

\*

```
* https://www.apache.org/licenses/LICENSE-2.0
```

\*

```
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
```

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/State.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2020 The Netty Project

\*

```
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
```

\*

```
* https://www.apache.org/licenses/LICENSE-2.0
```

\*

```
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
```

\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/PcapWriter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/EthernetPacket.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/PcapHeaders.java  
\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/PcapWriteHandler.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/UDPPacket.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/IPPacket.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/package-info.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/pcap/TCPpacket.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2018 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslSession.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterial.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialProvider.java  
\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslX509TrustManagerWrapper.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslCachingKeyMaterialProvider.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslPrivateKey.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SignatureAlgorithmConverter.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/DefaultOpenSslKeyMaterial.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslX509KeyManagerFactory.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/ExtendedOpenSslSession.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/OpenSslCachingX509KeyManagerFactory.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2020 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslClosedEngineException.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/address/ResolveAddressHandler.java  
\*  
/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ssl/SslHandshakeTimeoutException.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/logging/ByteBufFormat.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-jar/io/netty/handler/ipfilter/IpSubnetFilterRuleComparator.java  
\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/util/FingerprintTrustManagerFactoryBuilder.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ipfilter/IpSubnetFilter.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2021 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/BouncyCastleAlpnSslUtils.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/Ciphers.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslSessionCache.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslSessionId.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/BouncyCastleAlpnSslEngine.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/GroupsConverter.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/BouncyCastle.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslAsyncPrivateKeyMethod.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslClientSessionCache.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/AsyncRunnable.java

\*

/opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/OpenSslContextOption.java

\* /opt/cola/permits/1620675508\_1680116263.024557/0/netty-handler-4-1-90-final-sources-

jar/io/netty/handler/ssl/SslProtocols.java

```
* /opt/cola/permits/1620675508_1680116263.024557/0/netty-handler-4-1-90-final-sources-
jar/io/netty/handler/ssl/SslContextOption.java
```

# 1.130 proto-google-common-protos 2.11.0

## 1.130.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
```

Found in path(s):

```
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/rpc/error_details.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/fraction.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/cloud/extended_operations.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/month.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/service.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/logging.proto
*
/opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/error_reason.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/quota.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/logging/type/http_request.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/money.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/geo/type/viewport.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/rpc/status.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/log.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/latlng.proto
*
/opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/phone_number.proto
```



\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/longrunning/operations.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/visibility.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/rpc/code.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/type/postal\_address.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/auth.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/documentation.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/usage.proto  
\*  
/opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/type/color.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/context.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/routing.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/type/timeofday.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/type/interval.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/billing.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/monitoring.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/client.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/metric.proto  
\*  
/opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/rpc/context/attribute\_context.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/type/expr.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/config\_change.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/cloud/audit/audit\_log.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/source\_info.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/field\_behavior.proto  
\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/logging/type/log\_severity.proto

```
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/datetime.proto
*
/opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/control.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/dayofweek.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/monitored_resource.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/resource.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/localized_text.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/annotations.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/backend.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/distribution.proto
*
/opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/date.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/httpbody.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/consumer.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/quaternion.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/endpoint.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/http.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/system_parameter.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/api/label.proto
* /opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/calendar_period.proto
*
/opt/cola/permits/1620675467_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-
jar/google/type/decimal.proto
No license file was found, but licenses were detected in source scan.
```

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
```

// agreement (which includes confidentiality provisions). These features may

Found in path(s):

\* /opt/cola/permits/1620675467\_1680234968.5745006/0/proto-google-common-protos-2-11-0-1-jar/google/api/launch\_stage.proto

# 1.131 apache-client 2.19.2

## 1.131.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.132 profiles 2.19.2

### 1.132.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>



## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.133 regions 2.19.2

### 1.133.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify



the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[ ]"  
replaced with your own identifying information. (Don't include  
the brackets!) The text should be enclosed in the appropriate  
comment syntax for the file format. We also recommend that a  
file or class name and description of purpose be included on the  
same "printed page" as the copyright notice for easier  
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.134 metrics-spi 2.19.2

### 1.134.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.135 netty-codec-http 4.1.90.Final

## 1.135.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2019 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionFilter.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerProtocolHandler.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/CloseWebSocketFrame.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerProtocolHandshakeHandler.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshakerFactory.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker13.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker07.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker00.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/CorruptedWebSocketFrameException.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolConfig.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketDecoderConfig.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionFilterProvider.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketCloseStatus.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket00FrameDecoder.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker08.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerProtocolConfig.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project



\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/MixedFileUpload.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpHeaders.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpHeaderDateFormat.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultHttpResponse.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpContentEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultHttpObject.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultHttpHeaders.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpVersion.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpResponseEncoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspResponseEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/DefaultHttpDataFactory.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpContentDecompressor.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/ContinuationWebSocketFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/package-info.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/DiskFileUpload.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

jar/io/netty/handler/codec/http/QueryStringEncoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpResponseStatus.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/MemoryAttribute.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/spdy/SpdyHttpEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/TextWebSocketFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/DefaultHttpRequest.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker13.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/HttpPostBodyUtil.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpRequestEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/LastHttpContent.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/HttpData.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpRequest.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/spdy/DefaultSpdySettingsFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/InterfaceHttpData.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/MemoryFileUpload.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpResponse.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/AbstractDiskHttpData.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketHandshakeException.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/HttpDataFactory.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpObjectAggregator.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/spdy/SpdySession.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/CaseIgnoringComparator.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

jar/io/netty/handler/codec/http/multipart/Attribute.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/package-info.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketUtil.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpRequestDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker07.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/BinaryWebSocketFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpClientCodec.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/Cookie.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/DiskAttribute.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpResponseDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpObject.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpMessage.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/HttpPostRequestEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/rtsp/RtspResponseDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/FileUpload.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/CookieDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpContent.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/PongWebSocketFrame.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/rtsp/RtspRequestEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/QueryStringDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/package-info.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/MixedAttribute.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/package-info.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspRequestDecoder.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket00FrameEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultHttpContent.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspHeaders.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultHttpMessage.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspObjectDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspObjectEncoder.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspVersions.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpMethod.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/PingWebSocketFrame.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpServerCodec.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostStandardRequestDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpObjectDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/AbstractMemoryHttpData.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostRequestDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketVersion.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpObjectEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpResponseStreamIdHandler.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpConstants.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostMultipartRequestDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

```
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker00.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/rtsp/RtspMethods.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/DefaultCookie.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker08.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpContentDecoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/multipart/AbstractHttpData.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpHeaders.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/multipart/InterfaceHttpPostRequestDecoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshakerFactory.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/rtsp/package-info.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpContentCompressor.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/rtsp/RtspResponseStatuses.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/DefaultLastHttpContent.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/multipart/InternalAttribute.java
No license file was found, but licenses were detected in source scan.
```

```
The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

```
Found in path(s):
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/META-
INF/native-image/io.netty/netty-codec-http/native-image.properties
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*
```

```
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/cors/CorsHandler.java
```

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/cors/CorsConfig.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketChunkedInput.java
```

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/multipart/FileUploadUtil.java
```

```
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpServerKeepAliveHandler.java
```

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
```

~  
~ <https://www.apache.org/licenses/LICENSE-2.0>  
~  
~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.  
-->

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-codec-http/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameDecoderDelegate.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpHeaderValues.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/ServerCookieEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/package-info.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateFrameClientExtensionHandshaker.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/package-info.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

```

jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateClientExtensionHandshaker.j
ava
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionUtil.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtensionHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyHeaderBlockRawDecoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateDecoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/rtsp/RtspHeaderValues.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateEncoder.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtensionHandshaker.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/WebSocketServerCompressionHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/WebSocketClientCompressionHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/ClientCookieEncoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionDecoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerFrameDeflateDecoder.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateFrameServerExtensionHandshaker.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/Utf8FrameValidator.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/rtsp/RtspHeaderNames.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpMessageUtil.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpHeaderNames.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateEncoder.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtensionHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionEncoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyProtocolException.java

```



\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtension.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerFrameDeflateEncoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtensionHandshaker.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpStatusClass.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpHeadersEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameCodec.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockZlibDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateDecoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtension.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateServerExtensionHandshaker.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtension.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpChunkedInput.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionData.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/ClientCookieDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpExpectationFailedEvent.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/EmptyHttpHeaders.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/CookieEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/ServerCookieDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/CookieHeaderNames.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/CookieUtil.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/CombinedHttpHeaders.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpScheme.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/HttpUtil.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/DefaultCookie.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/package-info.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/ServerCookieEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspEncoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/Cookie.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/ClientCookieEncoder.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/rtsp/RtspDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/CookieDecoder.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/cookie/CookieUtil.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2021 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpMessageDecoderResult.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/CompressionEncoderFactory.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2019 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

/\*  
\* Adaptation of <https://bjoern.hoehrmann.de/utf-8/decoder/dfa/>  
\*  
\* Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software  
\* and associated documentation files (the "Software"), to  
deal in the Software without restriction,  
\* including without limitation the rights to use, copy, modify, merge, publish, distribute,  
\* sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is  
\* furnished to do so, subject to the following conditions:

\*  
\* The above copyright notice and this permission notice shall be included in all copies or  
\* substantial portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING  
\* BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR  
PURPOSE AND  
\* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE  
FOR ANY CLAIM,  
\* DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,  
ARISING FROM,  
\* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/Utf8Validator.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2022 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/multipart/AbstractMixedHttpData.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/TooLongHttpHeaderException.java

\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpHeaderValidationUtil.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/TooLongHttpContentException.java

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

jar/io/netty/handler/codec/http/TooLongHttpLineException.java  
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpClientUpgradeHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/HttpServerUpgradeHandler.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version
 * 2.0 (the "License"); you may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/cors/CorsConfigBuilder.java
No license file was found, but licenses were detected in source scan.
```

```
/*
```

\* Copyright 2017 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/HttpServerExpectContinueHandler.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/ReadOnlyHttpHeaders.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/http/websocketx/WebSocketScheme.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2013 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/spdy/SpdyGoAwayFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-  
jar/io/netty/handler/codec/spdy/DefaultSpdySynReplyFrame.java  
\*

/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyHeadersFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyDataFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdySynStreamFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyRstStreamFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeadersFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyDataFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyStreamStatus.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandler.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyPingFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockRawEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/FullHttpRequest.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyPingFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandshakeHandler.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockEncoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyWindowUpdateFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyGoAwayFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/DefaultFullHttpRequest.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpCodec.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketFrameEncoder.java  
\*  
/opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/SpdyRstStreamFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyWindowUpdateFrame.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-

```

jar/io/netty/handler/codec/spdy/SpdyHeaderBlockJZlibEncoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdySynReplyFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketFrameAggregator.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketProtocolHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/FullHttpResponse.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/DefaultSpdyStreamFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdySynStreamFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/DefaultFullHttpResponse.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdySettingsFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/FullHttpRequest.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketFrameDecoder.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdySessionHandler.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdySessionStatus.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyHeaderBlockZlibEncoder.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/DefaultSpdyHeaders.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/cors/package-info.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyVersion.java
*
/opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyCodecUtil.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyStreamFrame.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/ComposedLastHttpContent.java
* /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/spdy/SpdyHeaders.java

```

No license file was found, but licenses were detected in source scan.



```
/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
 * /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshakeException.java
 * /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshakeException.java
 *
 /opt/cola/permits/1620675358_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-
jar/io/netty/handler/codec/http/multipart/DeleteFileOnExitHook.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

```
// (BSD License: https://www.opensource.org/licenses/bsd-license)
// All rights reserved.
// Redistribution and use in source and binary forms, with or
// * Redistributions of source code must retain the above
// copyright notice, this list of conditions and the
// following disclaimer.
// * Redistributions
```

in binary form must reproduce the above  
// following disclaimer in the documentation and/or other  
// \* Neither the name of the Webbit nor the names of

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameDecoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameDecoder.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

// (BSD License: <https://www.opensource.org/licenses/bsd-license>)

// All rights reserved.

// Redistribution and use in source and binary forms, with or

// \* Redistributions of source code must retain the above

// copyright notice, this list of conditions and the

// following disclaimer.

// \* Redistributions

in binary form must reproduce the above

// following disclaimer in the documentation and/or other

// \* Neither the name of the Webbit nor the names of

Found in path(s):

\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameEncoder.java  
\* /opt/cola/permits/1620675358\_1680234971.5069332/0/netty-codec-http-4-1-90-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameEncoder.java

# 1.136 third-party-jackson-dataformat-cbor

## 2.19.2

### 1.136.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

# 1.137 aws-json-protocol 2.19.2

## 1.137.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.



"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.138 aws-core 2.19.2

### 1.138.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the



origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed  
with the accompanying software if such terms are included in the directory containing the accompanying software.  
Such other license terms will then apply in lieu of the terms of the software license above.

## 1.139 netty-transport 4.1.90.Final

### 1.139.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/StacklessClosedChannelException.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/socket/DuplexChannelConfig.java
```

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

/\*\*

\* Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in

\* its { @link ChannelPipeline }.

\*

\* <h3>Sub-types</h3>

\* <p>

\* { @link ChannelHandler } itself does not provide many methods, but you usually have to implement one of its subtypes:

\* <ul>

\* <li>{ @link

ChannelInboundHandler } to handle inbound I/O events, and</li>

\* <li>{ @link ChannelOutboundHandler } to handle outbound I/O operations.</li>

\* </ul>

\* </p>

\* <p>

\* Alternatively, the following adapter classes are provided for your convenience:

\* <ul>

\* <li>{ @link ChannelInboundHandlerAdapter } to handle inbound I/O events,</li>

\* <li>{ @link ChannelOutboundHandlerAdapter } to handle outbound I/O operations, and</li>

\* <li>{ @link ChannelDuplexHandler } to handle both inbound and outbound events</li>

\* </ul>

\* </p>

\* <p>

\* For more information, please refer to the documentation of each subtype.

\* </p>

\*

\* <h3>The context object</h3>

\* <p>

\* A { @link ChannelHandler } is provided with a { @link ChannelHandlerContext }

\* object. A { @link ChannelHandler } is supposed to interact with the

\* { @link ChannelPipeline } it belongs to via a context object. Using the

\* context object, the { @link ChannelHandler } can pass events upstream or downstream, modify the pipeline dynamically, or store the information (using { @link AttributeKey }s) which is specific to the handler.

\*

\* <h3>State management</h3>

\*

\* A { @link ChannelHandler } often needs to store some stateful information.

\* The simplest and recommended approach is to use member variables:

\* <pre>

\* public interface Message {

\* // your methods here

\* }

\*

\* public class DataServerHandler extends { @link SimpleChannelInboundHandler } &lt; Message &gt; {

\*

\* <b>private boolean loggedIn;</b>

\*

\* { @code @Override }

\* public void channelRead0({ @link ChannelHandlerContext } ctx, Message message) {

\* if (message instanceof LoginMessage) {

\* authenticate((LoginMessage) message);

\* <b>loggedIn = true;</b>

\* } else (message instanceof GetDataMessage) {

\* if (<b>loggedIn</b>) {

\* ctx.writeAndFlush(fetchSecret((GetDataMessage) message));

\* } else {

\* fail();

\* }

\*

\* }

\* }

\* ...

\* }

\* </pre>

\* Because the handler instance has a state variable which is dedicated to

\* one connection, you have to create a new handler instance for each new

\* channel to avoid a race condition where an unauthenticated client can get

\* the confidential information:

\* <pre>

\* // Create a new handler instance per channel.

\* // See { @link ChannelInitializer#initChannel(Channel)}.

\* public class DataServerInitializer extends { @link ChannelInitializer } &lt; { @link Channel } &gt; {

\* { @code @Override }

\* public void initChannel({ @link Channel } channel) {

\* channel.pipeline().addLast("handler", <b>new DataServerHandler()</b>);

\* }

\* }

```

*
* </pre>
*
* <h4>Using { @link AttributeKey}s</h4>
*
* Although it's recommended to use member variables to store the state of a
* handler, for some reason you might not want to create many handler instances.
* In such a case, you can use { @link AttributeKey}s which is provided by
* { @link ChannelHandlerContext}:
* <pre>
* public interface Message {
* // your methods here
* }
*
* { @code @Sharable}
* public class DataServerHandler extends { @link SimpleChannelInboundHandler}<Message> {
* private final { @link AttributeKey}<Boolean> auth =
* { @link AttributeKey#valueOf(String) AttributeKey.valueOf("auth")};
*
* { @code @Override}
* public void channelRead({ @link ChannelHandlerContext} ctx, Message message) {
* { @link Attribute}<Boolean> attr = ctx.attr(auth);
* if (message instanceof LoginMessage) {
* authenticate((LoginMessage) o);
* attr.set(true);
* } else (message instanceof GetDataMessage) {
* if (Boolean.TRUE.equals(attr.get())) {
* ctx.writeAndFlush(fetchSecret((GetDataMessage) o));
* } else {
* fail();
* }
* }
* }
* }
* ...
* }
* </pre>

```

Now that the state of the handler is attached to the { @link ChannelHandlerContext}, you can add the same handler instance to different pipelines:

```

* <pre>
* public class DataServerInitializer extends { @link ChannelInitializer}<Channel> {
*
* private static final DataServerHandler SHARED = new DataServerHandler();
*
* { @code @Override}
* public void initChannel({ @link Channel} channel) {
* channel.pipeline().addLast("handler", SHARED);
* }
* }
* </pre>

```

```

* }
* </pre>
*
*
* <h4>The { @code @Sharable } annotation</h4>
* <p>
* In the example above which used an { @link AttributeKey },
* you might have noticed the { @code @Sharable } annotation.
* <p>
* If a { @link ChannelHandler } is annotated with the { @code @Sharable }
* annotation, it means you can create an instance of the handler just once and
* add it to one or more { @link ChannelPipeline }s multiple times without
* a race condition.
* <p>
* If this annotation is not specified, you have to create
 a new handler
* instance every time you add it to a pipeline because it has unshared state
* such as member variables.
* <p>
* This annotation is provided for documentation purpose, just like
* the JCIP annotations.
*
* <h3>Additional resources worth reading</h3>
* <p>
* Please refer to the { @link ChannelHandler }, and
* { @link ChannelPipeline } to find out more about inbound and outbound operations,
* what fundamental differences they have, how they flow in a pipeline, and how to handle
* the operation in your application.
*/

```

Found in path(s):

```

* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelHandler.java

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

* Copyright 2013 The Netty Project

```

```

*

```

```

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

```

```

*

```

```

* https://www.apache.org/licenses/LICENSE-2.0

```

```

*

```

```

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations

```

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/oio/OioByteStreamChannel.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2017 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownReadComplete.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ChannelOutputShutdownEvent.java

\*

/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/DelegatingChannelPromiseNotifier.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/internal/ChannelUtils.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/FailedChannel.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/oio/OioDatagramChannelConfig.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/internal/package-info.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ChannelOutputShutdownException.java

\*

/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/oio/DefaultOioDatagramChannelConfig.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-

jar/io/netty/channel/PendingBytesTracker.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/DuplexChannel.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/BootstrapConfig.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelInboundInvoker.java

\*

/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/ServerBootstrapConfig.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/SelectStrategyFactory.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/AbstractBootstrapConfig.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/WriteBufferWaterMark.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/group/VoidChannelGroupFuture.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelOutboundInvoker.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/PreferHeapByteBufAllocator.java

\*

/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/SelectStrategy.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/DefaultSelectStrategyFactory.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/DefaultSelectStrategy.java



No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelHandlerAdapter.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/socket/oio/DefaultOioSocketChannelConfig.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/SimpleChannelInboundHandler.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelId.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelProgressiveFutureListener.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultChannelId.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/MessageSizeEstimator.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ConnectTimeoutException.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelProgressiveFuture.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultAddressedEnvelope.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/group/ChannelGroupException.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/socket/oio/OioServerSocketChannelConfig.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/socket/oio/DefaultOioServerSocketChannelConfig.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
```

```
jar/io/netty/channel/nio/SelectedSelectionKeySet.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/socket/oio/OioSocketChannelConfig.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/bootstrap/ChannelFactory.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultMessageSizeEstimator.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelOutboundBuffer.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/ChannelProgressivePromise.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultChannelProgressivePromise.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/AddressedEnvelope.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/group/ChannelGroup.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/group/DefaultChannelGroup.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/AbstractEventLoopGroup.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/group/ChannelMatcher.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/group/ChannelMatchers.java
No license file was found, but licenses were detected in source scan.
```

```
<!--
~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/META-
INF/maven/io.netty/netty-transport/pom.xml
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/SimpleChannelPool.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/MaxBytesRecvByteBufAllocator.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/ChannelPoolMap.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/ChannelPool.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/AbstractChannelPoolHandler.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/ChannelHealthChecker.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/MaxMessagesRecvByteBufAllocator.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultMaxBytesRecvByteBufAllocator.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/FixedChannelPool.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/AbstractChannelPoolMap.java
*
/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/package-info.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/pool/ChannelPoolHandler.java
* /opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultMaxMessagesRecvByteBufAllocator.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-
jar/io/netty/channel/DefaultChannelHandlerContext.java
```

No license file was found, but licenses were detected in source scan.

```
The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

Found in path(s):

```
*/opt/cola/permits/1620675365_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/META-
INF/native-image/io.netty.netty-transport/native-image.properties
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/package-info.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelConfig.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/ServerBootstrap.java
- \*
- /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelPipeline.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelInitializer.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/FailedChannelFuture.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/nio/package-info.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/embedded/EmbeddedEventLoop.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/oio/AbstractOioChannel.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/package-info.java
- \*
- /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/VoidChannelPromise.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/oio/AbstractOioMessageChannel.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/CompleteChannelFuture.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/AdaptiveRecvByteBufAllocator.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelInboundHandler.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ServerSocketChannelConfig.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelFuture.java
- \*
- /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/group/ChannelGroupFuture.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/FixedRecvByteBufAllocator.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ServerSocketChannel.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/nio/AbstractNioByteChannel.java
- \* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-

jar/io/netty/channel/SucceededChannelFuture.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelMetadata.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelOutboundHandler.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/oio/OioSocketChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/MultithreadEventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/DefaultEventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelInboundHandlerAdapter.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelFutureListener.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ThreadPerChannelEventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/AbstractChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/bootstrap/Bootstrap.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/package-info.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelOutboundHandlerAdapter.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/local/LocalEventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelException.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/DefaultChannelPromise.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/DefaultChannelPipeline.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/AbstractChannelHandlerContext.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelDuplexHandler.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ServerChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/embedded/package-info.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/embedded/EmbeddedSocketAddress.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/oio/package-info.java

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/local/LocalServerChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/DatagramChannelConfig.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/CombinedChannelDuplexHandler.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/SocketChannelConfig.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/oio/AbstractOioByteChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/EventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/oio/OioServerSocketChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/local/LocalChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelPromise.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelOption.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/DefaultSocketChannelConfig.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/DefaultChannelConfig.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/AbstractServerChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelPipelineException.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/oio/package-info.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/nio/NioEventLoopGroup.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/RecvByteBufAllocator.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/EventLoopException.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/group/package-info.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/embedded/EmbeddedChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/nio/NioServerSocketChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-

jar/io/netty/channel/EventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/nio/NioDatagramChannelConfig.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelPromiseAggregator.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/group/CombinedIterator.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/DefaultServerSocketChannelConfig.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/Channel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/nio/NioEventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/group/DefaultChannelGroupFuture.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/SocketChannel.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/oio/OioEventLoopGroup.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/DatagramChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/DatagramPacket.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/nio/ProtocolFamilyConverter.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/nio/AbstractNioChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelHandlerContext.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/local/LocalAddress.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/SingleThreadEventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/nio/NioSocketChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/socket/oio/OioDatagramChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/DefaultFileRegion.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/local/LocalChannelRegistry.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/nio/AbstractNioMessageChannel.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/FileRegion.java



\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/nio/NioTask.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/bootstrap/AbstractBootstrap.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownEvent.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/DefaultEventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/local/package-info.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/group/ChannelGroupFutureListener.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/nio/package-info.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/InternetProtocolFamily.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/DefaultDatagramChannelConfig.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelPromiseNotifier.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelFlushPromiseNotifier.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/PendingWriteQueue.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-

jar/io/netty/channel/AbstractEventLoop.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ReflectiveChannelFactory.java  
\*  
/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/embedded/EmbeddedChannelId.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ChannelFactory.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2021 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-  
jar/io/netty/channel/ServerChannelRecvByteBufAllocator.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2018 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/SimpleUserEventChannelHandler.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/nio/NioChannelOption.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2019 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ChannelHandlerMask.java  
\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/ExtendedClosedChannelException.java

\*

/opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/EventLoopTaskQueueFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2017 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the  
\* "License"); you may not use this file except in compliance with the License. You may obtain a  
\* copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/AbstractCoalescingBufferQueue.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/CoalescingBufferQueue.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2022 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675365\_1680116232.4282322/0/netty-transport-4-1-90-final-sources-jar/io/netty/channel/socket/nio/SelectorProviderUtil.java

# 1.140 auth 2.19.2

## 1.140.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,



whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.141 netty 4.1.90.Final

## 1.141.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/NetUtilInitializations.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/NetUtilSubstitutions.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/DomainWildcardMappingBuilder.java
```

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
 <modelVersion>4.0.0</modelVersion>
```

```
 <artifactId>jctools-core</artifactId>
 <groupId>org.jctools</groupId>
 <version>3.1.0</version>
 <name>Java Concurrency Tools Core Library</name>
 <description>Java Concurrency Tools Core Library</description>
 <packaging>bundle</packaging>
```

```

<dependencies>
 <dependency>
 <groupId>org.hamcrest</groupId>
 <artifactId>hamcrest-all</artifactId>
 <version>${hamcrest.version}</version>
 <scope>test</scope>
 </dependency>

 <dependency>
 <groupId>junit</groupId>
 <artifactId>junit</artifactId>
 <version>${junit.version}</version>
 <scope>test</scope>
 </dependency>

 <dependency>
 <groupId>com.google.guava</groupId>
 <artifactId>guava-testlib</artifactId>
 <version>${guava-testlib.version}</version>
 <scope>test</scope>
 </dependency>
</dependencies>
<build>
 <plugins>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-surefire-plugin</artifactId>
 <version>3.0.0-M3</version>
 <configuration>
 <includes>
 <include>*</include>
 </includes>
 </configuration>
 </plugin>
 <plugin>
 <groupId>org.apache.felix</groupId>
 <artifactId>maven-bundle-plugin</artifactId>
 <version>4.2.1</version>
 <extensions>>true</extensions>
 <configuration>
 <instructions>
 <Import-Package>sun.misc;resolution:=optional</Import-Package>
 </instructions>
 </configuration>
 </plugin>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-source-plugin</artifactId>

```

```

<version>3.2.0</version>
<executions>
 <execution>
 <id>attach-sources</id>
 <phase>verify</phase>
 <goals>
 <goal>jar-no-fork</goal>
 </goals>
 </execution>
</executions>
</plugin>
<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-javadoc-plugin</artifactId>
 <version>3.1.1</version>
 <configuration>
 <additionalOptions>
 <additionalOption>-Xdoclint:none</additionalOption>
 </additionalOptions>
 <source>8</source>
 </configuration>
 <executions>
 <execution>
 <id>attach-javadocs</id>
 <goals>
 <goal>jar</goal>
 </goals>
 </execution>
 </executions>
</plugin>
</plugins>
</build>

<distributionManagement>
 <repository>
 <id>bintray-jctools-jctools</id>
 <name>jctools-jctools</name>
 <url>https://api.bintray.com/maven/jctools/jctools/jctools-core/?publish=1</url>
 </repository>
</distributionManagement>

<url>https://github.com/JCTools</url>
<inceptionYear>2013</inceptionYear>

<licenses>
 <license>
 <name>Apache

```

```
License, Version 2.0</name>
 <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
 <distribution>repo</distribution>
</license>
</licenses>

<scm>
 <url>https://github.com/JCTools/JCTools</url>
 <connection>scm:git:https://github.com/JCTools/JCTools</connection>
 <tag>HEAD</tag>
</scm>

<developers>
 <developer>
 <url>https://github.com/nitsanw</url>
 </developer>
 <developer>
 <url>https://github.com/mjpt777</url>
 </developer>
 <developer>
 <url>https://github.com/RichardWarburton</url>
 </developer>
 <developer>
 <url>https://github.com/kay</url>
 </developer>
 <developer>
 <url>https://github.com/franz1981</url>
 </developer>
</developers>

<prerequisites>
 <maven>3.5.0</maven>
</prerequisites>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <java.version>1.6</java.version>
 <java.test.version>1.8</java.test.version>

 <maven.compiler.source>${java.version}</maven.compiler.source>
 <maven.compiler.target>${java.version}</maven.compiler.target>
 <maven.compiler.testSource>${java.test.version}</maven.compiler.testSource>
 <maven.compiler.testTarget>${java.test.version}</maven.compiler.testTarget>

 <hamcrest.version>1.3</hamcrest.version>
 <junit.version>4.12</junit.version>
 <guava-testlib.version>21.0</guava-testlib.version>
</properties>
```

</project>

Found

in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/META-INF/maven/org.jctools/jctools-core/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the

\* "License"); you may not use this file except in compliance with the License. You may obtain a

\* copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/ByteProcessor.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/MathUtil.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

```

* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/DefaultFutureListeners.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/Future.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ResourceLeakDetector.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/ConcurrentSet.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/ScheduledFutureTask.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/TypeParameterMatcher.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ReferenceCountUtil.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ConstantPool.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/ReadOnlyIterator.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/RecyclableArrayList.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/GenericFutureListener.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/ProgressivePromise.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/EmptyArrays.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/DefaultThreadFactory.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/DefaultPromise.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/package-info.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/CompleteFuture.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/Recycler.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ReferenceCounted.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/DefaultProgressivePromise.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/ImmediateExecutor.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-

```

jar/io/netty/util/ResourceLeak.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/ImmediateEventExecutor.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/PromiseTask.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/PlatformDependent0.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/ProgressiveFuture.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/AbstractEventExecutorGroup.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/AbstractEventExecutor.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/AbstractReferenceCounted.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/GenericProgressiveFutureListener.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/FutureListener.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/PendingWrite.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/ScheduledFuture.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/package-info.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/NoOpTypeParameterMatcher.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/ThreadPerTaskExecutor.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/Version.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/Promise.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/ResourceLeakException.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/AppendableCharSequence.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2017 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:



\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/NettyRuntime.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ReflectionUtil.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/LongAdderCounter.java  
\*  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/EmptyPriorityQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/PriorityQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/UncheckedBooleanSupplier.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/LocationAwareSlf4JLogger.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ObjectCleaner.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/SuppressForbidden.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2018 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/SuppressJava6Requirement.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/PromiseNotificationUtil.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/Log4J2LoggerFactory.java

\*

/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/PromiseCombiner.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/OrderedEventExecutor.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/EventExecutorChooserFactory.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ThrowableUtil.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/UnstableApi.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/RejectedExecutionHandlers.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/DomainNameMappingBuilder.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/Log4J2Logger.java

\*

/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/UnaryPromiseNotifier.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/ResourceLeakTracker.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/ResourceLeakDetectorFactory.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/IntSupplier.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ConstantTimeUtils.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutorChooserFactory.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/OutOfDirectMemoryError.java  
\*

/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/NativeLibraryUtil.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/RejectedExecutionHandler.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/SocketUtils.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/UnorderedThreadPoolEventExecutor.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/BooleanSupplier.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/NonStickyEventExecutorGroup.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/MacAddressUtil.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/LongCounter.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/AbstractScheduledEventExecutor.java

\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/DefaultPriorityQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/ThreadProperties.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/HashingStrategy.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/DomainMappingBuilder.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/PriorityQueueNode.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/AsyncMapping.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2017 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/Cleaner.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/CleanerJava9.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/FastThreadLocalRunnable.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2019 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>

\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/svm/CleanerJava6Substitution.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/svm/package-info.java

\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/Hidden.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/svm/PlatformDependentSubstitution.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/svm/UnsafeRefArrayAccessSubstitution.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/svm/PlatformDependent0Substitution.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/ObjectPool.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/ReferenceCountUpdater.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/ThreadExecutorMap.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-

```
jar/io/netty/util/concurrent/FastThreadLocal.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/UnpaddedInternalThreadLocalMap.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/DomainNameMapping.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ThreadDeathWatcher.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/PromiseNotifier.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/Mapping.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/AsciiString.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/InternalThreadLocalMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/ResourceLeakHint.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/NativeLibraryLoader.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/concurrent/PromiseAggregator.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/IntegerHolder.java
No license file was found, but licenses were detected in source scan.
```

```
The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

Found in path(s):

```
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/META-INF/native-
image/io.netty.netty-common/native-image.properties
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright 2014 The Netty Project
```

```
*
```

```
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
```

```
*
```

```
* https://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/  
/\*  
\* Written by Doug Lea with assistance from members of JCP JSR-166  
\* Expert Group and released to the public domain, as explained at  
\* <https://creativecommons.org/publicdomain/zero/1.0/>  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/ThreadLocalRandom.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/MpscGrowableArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/QueueFactory.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/AtomicReferenceArrayQueue.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/util/UnsafeLongArrayAccess.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/MpmcArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpmcAtomicArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpmcAtomicArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingHashSet.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-

```

jar/io/netty/util/internal/shaded/org/jctools/queues/MpmcUnboundedXaddArrayQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpscLinkedQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/util/UnsafeAccess.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpscUnboundedXaddArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/SpscArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscChunkedAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/util/Pow2.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/AbstractEntry.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingSetInt.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/ConcurrentSequencedCircularArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpscUnboundedArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/SpmcArrayQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingHashMapLong.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscUnboundedAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/package-info.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MessagePassingQueueUtil.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SequencedAtomicReferenceArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpscLinkedAtomicQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/BaseLinkedAtomicQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/BaseMpscLinkedAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-

```



```

jar/io/netty/util/internal/shaded/org/jctools/queues/SupportsIterator.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpSCUnboundedAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpSCGrowableAtomicArrayQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/ConcurrentCircularArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/BaseLinkedListQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/util/UnsafeRefArrayAccess.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/spec/Ordering.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpMCUnboundedXaddChunk.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpSCArrayQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/SpSCLinkedListQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/BaseSpSCLinkedListAtomicArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/AtomicQueueFactory.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpSCLinkedListAtomicQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/spec/Preference.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpSCChunkedArrayQueue.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/ConcurrentAutoTable.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MessagePassingQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/util/RangeUtil.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/SpSCChunkedArrayQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpSCCompoundQueue.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/queues/MpSCUnboundedXaddChunk.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingIdentityHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-

```

jar/io/netty/util/internal/shaded/org/jctools/queues/BaseMpscLinkedListArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/util/PortableJvmInfo.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/util/InternalAPI.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/spec/ConcurrentQueueSpec.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/IndexedQueueSizeUtil.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscGrowableAtomicArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/util/UnsafeJvmInfo.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/LinkedListAtomicNode.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/LinkedListQueueNode.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/MpscBlockingConsumerArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/BaseSpscLinkedListArrayQueue.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/SpscGrowableArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpscChunkedAtomicArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/SpscUnboundedArrayQueue.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpscAtomicArrayQueue.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/SucceededFuture.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/AbstractInternalLogger.java
- \*
- /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/JdkLoggerFactory.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/Log4JLoggerFactory.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/Slf4JLogger.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/Timer.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/CharsetUtil.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/GlobalEventExecutor.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/SingleThreadEventExecutor.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutorGroup.java
- \*
- /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/AbstractConstant.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/EventExecutor.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/InternalLogLevel.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/IllegalReferenceCountException.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutor.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/CommonsLoggerFactory.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/Signal.java
- \*
- /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/package-info.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/AttributeKey.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/Constant.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/NetUtil.java
- \* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-

jar/io/netty/util/Timeout.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/Attribute.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/FailedFuture.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/MultithreadEventExecutorGroup.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/DefaultAttributeMap.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/InternalLoggerFactory.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/Slf4JLoggerFactory.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/AttributeMap.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/PlatformDependent.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/SystemPropertyUtil.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/package-info.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/TimerTask.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/BlockingOperationException.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/concurrent/EventExecutorGroup.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/HashedWheelTimer.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/StringUtil.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.  
\*/  
/\*\*  
\* Copyright (c) 2004-2011 QOS.ch  
\* All rights reserved.  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining  
\* a copy of this software and associated documentation files (the  
\* "Software"), to deal in the Software without restriction, including  
\* without  
\* limitation the rights to use, copy, modify, merge, publish,  
\* distribute, sublicense, and/or sell copies of the Software, and to  
\* permit persons to whom the Software is furnished to do so, subject to  
\* the following conditions:  
\*  
\* The above copyright notice and this permission notice shall be  
\* included in all copies or substantial portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
\* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE  
\* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION  
\* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
\*  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/CommonsLogger.java  
\*  
/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/JdkLogger.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/InternalLogger.java  
\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-  
jar/io/netty/util/internal/logging/Log4JLogger.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2018 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the  
\* "License"); you may not use this file except in compliance with the License. You may obtain a  
\* copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>

\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/ByteProcessorUtils.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ResourcesUtil.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2021 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/ClassInitializerUtil.java

No license file was found, but licenses were detected in source scan.

# Copyright 2019 The Netty Project

#

# The Netty Project licenses this file to you under the Apache License,  
# version 2.0 (the "License"); you may not use this file except in compliance  
# with the License. You may obtain a copy of the License at:

#

# <https://www.apache.org/licenses/LICENSE-2.0>

#

# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
# WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
# License for the specific language governing permissions and limitations

# under the License.

io.netty.util.internal.Hidden\$NettyBlockHoundIntegration

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/META-INF/services/reactor.blockhound.integration.BlockHoundIntegration

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

/\*\*

\* Copyright (c) 2004-2011 QOS.ch

\* All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without

limitation the rights to use, copy, modify, merge, publish,

\* distribute, sublicense, and/or sell copies of the Software, and to

\* permit persons to whom the Software is furnished to do so, subject to

\* the following conditions:

\*

\* The above copyright notice and this permission notice shall be

\* included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

\* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

\* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION

\* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

\* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/FormattingTuple.java

\*

/opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/logging/MessageFormatter.java

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2012 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,  
~ version 2.0 (the "License"); you may not use this file except in compliance  
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.

-->

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-common/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the  
\* "License"); you may not use this file except in compliance with the License. You may obtain a  
\* copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-



```
jar/io/netty/util/collection/CharObjectMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/LongCollections.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/CharCollections.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/IntCollections.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ByteObjectMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/CharObjectHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ShortCollections.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/LongObjectHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/IntObjectHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/internal/ObjectUtil.java
*
/opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ShortObjectHashMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ShortObjectMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/IntObjectMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/LongObjectMap.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ByteCollections.java
* /opt/cola/permits/1620675283_1680116247.569084/0/netty-common-4-1-90-final-sources-
jar/io/netty/util/collection/ByteObjectHashMap.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
```

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/concurrent/FastThreadLocalThread.java

\* /opt/cola/permits/1620675283\_1680116247.569084/0/netty-common-4-1-90-final-sources-jar/io/netty/util/internal/CleanerJava6.java

# 1.142 endpoints-spi 2.19.2

## 1.142.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.143 netty-codec 4.1.90.Final

## 1.143.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2021 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/BrotliOptions.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/GzipOptions.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/BrotliDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/ZstdConstants.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
```

```
jar/io/netty/handler/codec/compression/Zstd.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/DeflateOptions.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/ZstdEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/ZstdOptions.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/BrotliEncoder.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/CompressionOptions.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Brotli.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/StandardCompressionOptions.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

```
Found in path(s):
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Lz4XXHash32.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
*/
```



\* Unless required by applicable law or agreed to in writing, software  
 \* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
 \* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
 \* License for the specific language governing permissions and limitations  
 \* under the License.

\*/  
 /\*\*

\* A decoder that splits the received {@link ByteBuf}s dynamically by the  
 \* value of the length field in the message. It is particularly useful when you  
 \* decode a binary message which has an integer header field that represents the  
 \* length of the message body or the whole message.

\*

<p>

\* {@link LengthFieldBasedFrameDecoder} has many configuration parameters so  
 \* that it can decode any message with a length field, which is often seen in  
 \* proprietary client-server protocols. Here are some example that will give  
 \* you the basic idea on which option does what.

\*

\* <h3>2 bytes length field at offset 0, do not strip header</h3>

\*

\* The value of the length field in this example is <tt>12 (0x0C)</tt> which  
 \* represents the length of "HELLO, WORLD". By default, the decoder assumes  
 \* that the length field represents the number of the bytes that follows the  
 \* length field. Therefore, it can be decoded with the simplistic parameter  
 \* combination.

\* <pre>

```
* lengthFieldOffset = 0
* lengthFieldLength = 2
* lengthAdjustment = 0
* initialBytesToStrip = 0 (= do not strip header)
```

\*

\* BEFORE DECODE (14 bytes)      AFTER DECODE (14 bytes)

```
* +-----+-----+ +-----+-----+
```

```
* | Length | Actual Content
```

```
|---->| Length | Actual Content |
```

```
* | 0x000C | "HELLO, WORLD" | | 0x000C | "HELLO, WORLD" |
```

```
* +-----+-----+ +-----+-----+
```

\* </pre>

\*

\* <h3>2 bytes length field at offset 0, strip header</h3>

\*

\* Because we can get the length of the content by calling  
 \* {@link ByteBuf#readableBytes()}, you might want to strip the length  
 \* field by specifying <tt>initialBytesToStrip</tt>. In this example, we  
 \* specified <tt>2</tt>, that is same with the length of the length field, to  
 \* strip the first two bytes.

\* <pre>

```
* lengthFieldOffset = 0
```

```

* lengthFieldLength = 2
* lengthAdjustment = 0
* initialBytesToStrip = 2 (= the length of the Length field)
*
* BEFORE DECODE (14 bytes) AFTER DECODE (12 bytes)
* +-----+-----+ +-----+
* | Length | Actual Content |---->| Actual Content |
* | 0x000C | "HELLO, WORLD" | | "HELLO, WORLD" |
* +-----+-----+ +-----+
* </pre>
*
* <h3>2 bytes length field at offset 0, do not strip header, the length field
* represents the length of the whole message</h3>
*
* In most cases, the length field represents the length of the message body
* only, as shown in the previous examples. However, in some protocols, the
* length field represents the length of the whole message, including the
* message header. In such a case, we specify a non-zero
* <tt>lengthAdjustment</tt>. Because the length value in this example message
* is always greater than the body length by <tt>2</tt>, we specify <tt>-2</tt>
* as <tt>lengthAdjustment</tt> for compensation.
* </pre>
* lengthFieldOffset = 0
* lengthFieldLength = 2
* lengthAdjustment = -2 (= the length of the Length field)
* initialBytesToStrip = 0
*
* BEFORE DECODE (14 bytes) AFTER DECODE (14 bytes)
* +-----+-----+ +-----+-----+
* | Length | Actual Content |---->| Length | Actual
 Content |
* | 0x000E | "HELLO, WORLD" | | 0x000E | "HELLO, WORLD" |
* +-----+-----+ +-----+-----+
* </pre>
*
* <h3>3 bytes length field at the end of 5 bytes header, do not strip header</h3>
*
* The following message is a simple variation of the first example. An extra
* header value is prepended to the message. <tt>lengthAdjustment</tt> is zero
* again because the decoder always takes the length of the prepended data into
* account during frame length calculation.
* </pre>
* lengthFieldOffset = 2 (= the length of Header 1)
* lengthFieldLength = 3
* lengthAdjustment = 0
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)

```

```

* +-----+-----+-----+ +-----+-----+-----+
* | Header 1 | Length | Actual Content |----->| Header 1 | Length | Actual Content |
* | 0xCAFE | 0x00000C | "HELLO, WORLD" |
* | 0xCAFE | 0x00000C | "HELLO, WORLD" |
* +-----+-----+-----+ +-----+-----+-----+
* </pre>
*
* <h3>3 bytes length field at the beginning of 5 bytes header, do not strip header</h3>
*
* This is an advanced example that shows the case where there is an extra
* header between the length field and the message body. You have to specify a
* positive <tt>lengthAdjustment</tt> so that the decoder counts the extra
* header into the frame length calculation.
* <pre>
* lengthFieldOffset = 0
* lengthFieldLength = 3
* lengthAdjustment = 2 (= the length of Header 1)
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +-----+-----+-----+ +-----+-----+-----+
* | Length | Header 1 | Actual Content |----->| Length | Header 1 | Actual Content |
* | 0x00000C | 0xCAFE | "HELLO, WORLD" | | 0x00000C | 0xCAFE | "HELLO,
* WORLD" |
* +-----+-----+-----+ +-----+-----+-----+
* </pre>
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
* strip the first header field and the length field</h3>
*
* This is a combination of all the examples above. There are the prepended
* header before the length field and the extra header after the length field.
* The prepended header affects the <tt>lengthFieldOffset</tt> and the extra
* header affects the <tt>lengthAdjustment</tt>. We also specified a non-zero
* <tt>initialBytesToStrip</tt> to strip the length field and the prepended
* header from the frame. If you don't want to strip the prepended header, you
* could specify <tt>0</tt> for <tt>initialBytesToSkip</tt>.
* <pre>
* lengthFieldOffset = 1 (= the length of HDR1)
* lengthFieldLength = 2
* lengthAdjustment = 1 (= the length of HDR2)
* initialBytesToStrip = 3 (= the length of HDR1 +
* LEN)
*
* BEFORE DECODE (16 bytes) AFTER DECODE (13 bytes)
* +-----+-----+-----+ +-----+-----+-----+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x000C | 0xFE | "HELLO, WORLD" | | 0xFE | "HELLO, WORLD" |

```

```

* +-----+-----+-----+-----+ +-----+-----+
* </pre>
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
* strip the first header field and the length field, the length field
* represents the length of the whole message</h3>
*
* Let's give another twist to the previous example. The only difference from
* the previous example is that the length field represents the length of the
* whole message instead of the message body, just like the third example.
* We have to count the length of HDR1 and Length into <tt>lengthAdjustment</tt>.
* Please note that we don't need to take the length of HDR2 into account
* because
the length field already includes the whole header length.
* <pre>
* lengthFieldOffset = 1
* lengthFieldLength = 2
* lengthAdjustment = -3 (= the length of HDR1 + LEN, negative)
* initialBytesToStrip = 3
*
* BEFORE DECODE (16 bytes) AFTER DECODE (13 bytes)
* +-----+-----+-----+-----+ +-----+-----+
* | HDR1 | Length | HDR2 | Actual Content |---->| HDR2 | Actual Content |
* | 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" | | 0xFE | "HELLO, WORLD" |
* +-----+-----+-----+-----+ +-----+-----+
* </pre>
* @see LengthFieldPrepender
*/

```

Found in path(s):

```

* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/LengthFieldBasedFrameDecoder.java

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

* Copyright 2012 The Netty Project

```

```

*

```

```

* The Netty Project licenses this file to you under the Apache License,

```

```

* version 2.0 (the "License"); you may not use this file except in compliance

```

```

* with the License. You may obtain a copy of the License at:

```

```

*

```

```

* https://www.apache.org/licenses/LICENSE-2.0

```

```

*

```

```

* Unless required by applicable law or agreed to in writing, software

```

```

* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

```

```

* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

```

```

* License for the specific language governing permissions and limitations

```

```

* under the License.

```

```
*/
/*
* Written by Robert Harder and released to the public domain, as explained at
* https://creativecommons.org/licenses/publicdomain
*/
/**
* Utility class for {@link ByteBuf} that encodes and decodes to and from
* Base64 notation.
*

* The encoding and decoding algorithm in this class has been derived from
* http://iharder.sourceforge.net/current/java/base64/ Robert Harder's Public Domain
* Base64 Encoder/Decoder.

*/
```

Found in path(s):

```
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/base64/Base64.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/DatagramPacketDecoder.java
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/CodecOutputList.java
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/DateFormatter.java
*
```

```
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/ByteBufChecksum.java
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/string/LineEncoder.java
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
```

jar/io/netty/handler/codec/DatagramPacketEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/compression/CompressionUtil.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/string/LineSeparator.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufEncoderNano.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufEncoder.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufVarint32LengthFieldPrepender.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/ProtocolDetectionResult.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/ProtocolDetectionState.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/HeadersUtils.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/UnsupportedValueConverter.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufDecoderNano.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/protobuf/ProtobufDecoder.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2023 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/EncoderUtil.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

```
/*
 * Written by Robert Harder and released to the public domain, as explained at
 * https://creativecommons.org/licenses/publicdomain
 */
```

```
/**
 * Enumeration of supported Base64 dialects.
 * <p>
 * The internal lookup tables in this class has been derived from
 * Robert
 * Harder's Public Domain
 * Base64 Encoder/Decoder.
 */
```

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/base64/Base64Dialect.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Crc32c.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/xml/XmlFrameDecoder.java

\*

/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/JdkZlibDecoder.java

No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,  
# version 2.0 (the "License"); you may not use this file except in compliance  
# with the License. You may obtain a copy of the License at:  
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/META-INF/native-image/io.netty/netty-codec/native-image.properties

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:



\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/LzmaFrameEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2MoveToFrontTable.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/json/package-info.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/SnappyFramedEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Crc32.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Lz4Constants.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/LzfEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockDecompressor.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2MTFAndRLE2StageEncoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/DecoderResultProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2DivSufSort.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/json/JsonObjectDecoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2Decoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockCompressor.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanAllocator.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageDecoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/AsciiHeadersEncoder.java

```
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/SnappyFramedDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2Encoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Lz4FrameDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/LzfDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/FastLzFrameDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Lz4FrameEncoder.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2Rand.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2HuffmanStageEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/FastLzFrameEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2BitWriter.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/FastLz.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/MessageAggregationException.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2BitReader.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/Bzip2Constants.java
No license file was found, but licenses were detected in source scan.
```

```
<!--
~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-codec/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/CachingClassResolver.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/MessageToMessageDecoder.java

\*

/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/Snappy.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/MessageAggregator.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/xml/package-info.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/ZlibWrapper.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/base64/package-info.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/SoftReferenceMap.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/MessageToMessageCodec.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/CorruptedFrameException.java

\*

/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/ByteToMessageDecoder.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/ClassResolvers.java

```

* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/PrematureChannelClosureException.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/MarshallingDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/CompatibleMarshallingDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/string/StringDecoder.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/SnappyFrameEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/serialization/ObjectEncoderOutputStream.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/MarshallingEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/CompatibleMarshallingEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/serialization/CompactObjectOutputStream.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/MarshallerProvider.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/ReplayingDecoder.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/bytes/package-info.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/base64/Base64Decoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/marshalling/ChannelBufferByteInput.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/serialization/ObjectEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/serialization/ObjectDecoderInputStream.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/DelimiterBasedFrameDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/serialization/package-info.java
*
/opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/compression/ZlibUtil.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/CodecException.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/LineBasedFrameDecoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-
jar/io/netty/handler/codec/bytes/ByteArrayEncoder.java
* /opt/cola/permits/1620675259_1680236261.0138025/0/netty-codec-4-1-90-final-sources-

```

jar/io/netty/handler/codec/DecoderResult.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/JdkZlibEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/DecoderException.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/ClassResolver.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/CompressionException.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/LengthFieldPrepender.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/CompactObjectInputStream.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/ZlibEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/LimitingByteInput.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/string/StringEncoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteOutput.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/MessageToByteEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/ClassLoaderClassResolver.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/DefaultUnmarshallerProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/ReferenceMap.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/Delimiters.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayDecoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/ContextBoundUnmarshallerProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/package-info.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/string/package-info.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/JZlibEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/ByteToMessageCodec.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameDecoder.java

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/ObjectDecoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/FixedLengthFrameDecoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/TooLongFrameException.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/ZlibDecoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/ZlibCodecFactory.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/UnmarshallerProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/UnsupportedMessageTypeException.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/package-info.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/WeakReferenceMap.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/DecompressionException.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/protobuf/package-info.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/MessageToMessageEncoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/JZlibDecoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalUnmarshallerProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/serialization/CompatibleObjectEncoder.java  
\*  
/opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/DefaultMarshallerProvider.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/EncoderException.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/ReplayingDecoderByteBuf.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/compression/package-info.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/base64/Base64Encoder.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalMarshallerProvider.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the  
\* "License"); you may not use this file except in compliance with the License. You may obtain a  
\* copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/DefaultHeaders.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/Headers.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/EmptyHeaders.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License, version 2.0 (the  
\* "License"); you may not use this file except in compliance with the License. You may obtain a  
\* copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/DefaultHeadersImpl.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/ValueConverter.java  
\* /opt/cola/permits/1620675259\_1680236261.0138025/0/netty-codec-4-1-90-final-sources-  
jar/io/netty/handler/codec/CharSequenceValueConverter.java

# 1.144 aws-query-protocol 2.19.2

## 1.144.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](https://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all



other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

# 1.145 netty-transport-native-unix-common

## 4.1.90.Final

### 1.145.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_buffer.c
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/PreferredDirectByteBufferAllocator.java
*
/opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_buffer.h
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/Buffer.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
```

\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_jni.h  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/UnixChannelUtil.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2015 The Netty Project  
\*  
\* The Netty Project licenses this file to you under the Apache License,  
\* version 2.0 (the "License"); you may not use this file except in compliance  
\* with the License. You may obtain a copy of the License at:  
\*  
\* <https://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_socket.c  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/Socket.java  
\*

/opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/FileDescriptor.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/DomainSocketAddress.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/NativeInetAddress.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/UnixChannel.java

```
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/ServerDomainSocketChannel.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainSocketChannelConfig.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_errors.c
*
/opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_errors.h
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/Errors.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DatagramSocketAddress.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_filedescriptor.h
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_socket.h
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainSocketReadMode.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainSocketChannel.java
*
/opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix_filedescriptor.c
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright 2020 The Netty Project
```

```
*
```

```
* The Netty Project licenses this file to you under the Apache License,
```

```
* version 2.0 (the "License"); you may not use this file except in compliance
```

```
* with the License. You may obtain a copy of the License at:
```

```
*
```

```
* https://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

```
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
```

```
* License for the specific language governing permissions and limitations
```

```
* under the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix.c
```

```
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/netty_unix.h
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2021 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/SegmentedDatagramPacket.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainDatagramChannelConfig.java
```

```
*
 /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainDatagramChannel.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainDatagramSocketAddress.java
* /opt/cola/permits/1620675307_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-
jar/io/netty/channel/unix/DomainDatagramPacket.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2022 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```



Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/IntegerUnixChannelOption.java

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/RawUnixChannelOption.java

\*

/opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/GenericUnixChannelOption.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/IovArray.java

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/package-info.java

\*

/opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/UnixChannelOption.java

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/Unix.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016 The Netty Project

\*

\* The Netty Project licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and limitations  
\* under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_limits.c  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_util.h  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java  
\*  
/opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/Limits.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/SocketWritableByteChannel.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/PeerCredentials.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_limits.h  
\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/netty\_unix\_util.c

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2016 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,  
~ version 2.0 (the "License"); you may not use this file except in compliance  
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.

-->

Found in path(s):

\* /opt/cola/permits/1620675307\_1680116248.8153183/0/netty-transport-native-unix-common-4-1-90-final-sources-jar/META-INF/maven/io.netty/netty-transport-native-unix-common/pom.xml

# 1.146 kinesis 2.19.2

## 1.146.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](https://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.147 dynamodb 2.19.2

### 1.147.1 Available under license :

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020  
The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,



and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding  
those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade  
names, trademarks, service marks, or product names of the Licensor,  
except as required for reasonable and customary use in describing the  
origin of the Work and reproducing the  
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or  
agreed to in writing, Licensor provides the Work (and each  
Contributor provides its Contributions) on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

## 1.148 opentelemetry-context 1.24.0

### 1.148.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The gRPC Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

/\*

\* Copyright 2020 LINE Corporation

\*

\* LINE Corporation licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by

applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/LazyStorage.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2020 LINE Corporation

\*

\* LINE Corporation licenses this file to you under the Apache License,

\* version 2.0 (the "License"); you may not use this file except in compliance

\* with the License. You may obtain a copy of the License at:

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

\* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and limitations

\* under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/ContextStorage.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015 The gRPC Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/ArrayBasedContext.java

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/Context.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright Rafael Winterhalter

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/internal/shaded/AbstractWeakConcurrentMap.java

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/internal/shaded/WeakConcurrentMap.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013-2020 The OpenZipkin Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677361\_1680116245.8199492/0/opentelemetry-context-1-24-0-sources-jar/io/opentelemetry/context/StrictContextStorage.java

# 1.149 aws-java-sdk-core 1.12.430

## 1.149.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) 2019. Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/ReflectionMethodInvoker.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/transform/EnhancedJsonErrorUnmarshaller.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not
 * use this file except in compliance with the License. A copy of the License is
 * located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/JsonFactory.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/transform/SimpleTypeJsonUnmarshallers.java
```



\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonParser.java

No license file was found, but licenses were detected in source scan.

/\*\*

\* Copyright 2016-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\* <p/>

\* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with

\* the License. A copy of the License is located at

\* <p/>

\* <http://aws.amazon.com/apache2.0>

\* <p/>

\* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR

\* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions

\* and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/Partitions.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2016-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/RegionMetadataFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/client/impl/CRC32ChecksumResponseInterceptor.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/Service.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/RegionImpl.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/LegacyRegionXmlLoadUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/Partition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkStructuredIonFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/CredentialScope.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/Endpoint.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/PartitionRegionImpl.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/SdkProxyRoutePlanner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/PartitionMetadataProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/AbstractRegionMetadataProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/InMemoryRegionsProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/LegacyRegionXmlMetadataBuilder.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/InMemoryRegionImpl.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/SimpleTypeCborUnmarshallers.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/PartitionsLoader.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/partitions/model/Region.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkIonGenerator.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/RegionMetadataProvider.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with

\* the License. A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*  
\* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR  
\* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions  
\* and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/GuardedBy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/Beta.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/SdkProtectedApi.java  
\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/NotThreadSafe.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/SdkInternalApi.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/ThreadSafe.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/SdkTestInternalApi.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/Immutable.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2020-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at

\*  
\* <http://aws.amazon.com/apache2.0>

\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/RegionFromEndpointResolverAwareSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/endpoint/RegionFromEndpointResolver.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/endpoint/DefaultRegionFromEndpointResolver.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/MetadataSupportedRegionFromEndpointProvider.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2011-2023 Amazon Technologies, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at:

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES

\* OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/ServiceClientHolderInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/json/Jackson.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/MetricAdmin.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/jmx/JmxInfoProviderSupport.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWSSessionCredentialsProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/jmx/spi/JmxInfoProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/RequestClientOptions.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/jmx/spi/SdkMBeanRegistry.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/MetricAdminMBean.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/BasicSessionCredentials.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/ClassLoaderHelper.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWSSessionCredentials.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/ClientConnectionRequestFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/config/HostRegexToRegionMapping.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/internal/ServiceMetricTypeGuesser.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/arn/AwsResource.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/CollectionUtils.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/StandardErrorUnmarshaller.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/AwsProfileEndpointDiscoveryProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/resources/package-info.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/ProgressListener.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/config/HostRegexToRegionMappingJsonHelper.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/HttpSuccessStatusAcceptor.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/EndpointDiscoveryIdentifiersRefreshCache.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/PollingStrategyContext.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ClientConfiguration.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/PropertiesCredentials.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/RequestHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/MetricType.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/RetryUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWS4UnsignedPayloadSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ResponseMetadata.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/HttpResponse.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/Response.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/package-info.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/ApiMonitoringEvent.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/AmazonWebServiceResponse.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/RepeatableInputStreamRequestEntity.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/Action.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/FixedDelayStrategy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterExecution.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/config/HttpClientConfigJsonHelper.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/EnvironmentVariableCsmConfigurationProvider.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/SimpleTypeJsonUnmarshallers.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/internal/AgentMonitoringListener.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/HandlerAfterAttemptContext.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/DefaultErrorResponseHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

jar/com/amazonaws/monitoring/internal/ClientSideMonitoringRequestHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/AbstractAWSSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/config/InternalConfig.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/MaxAttemptsRetryStrategy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/HttpFailureStatusAcceptor.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/SDKGlobalConfiguration.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkPredicate.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/endpointdiscovery/DaemonThreadFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/RuntimeHttpUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/cache/KeyConverter.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/metrics/ServiceLatencyProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/NoOpWaiterHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/AWSServiceMetrics.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/SystemPropertyTlsKeyManagersProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/ServiceNameFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/retry/internal/AuthRetryParameters.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/WaiterAcceptor.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/WaiterState.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/HttpMethod.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/conditions/NumericCondition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/monitoring/internal/AsynchronousAgentDispatcher.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/HttpResponseHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/CountingInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/apachehttpclientconfig.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/HandlerChainFactory.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/RequestHandler2.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/impl/client/SdkHttpRequestRetryHandler.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWS3Signer.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/WebIdentityTokenCredentialsProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/AbstractRequestHandler.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/EndpointDiscoveryRefreshCache.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/ByteThroughputHelper.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/MetricCollector.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/CsmConfiguration.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/AmazonHttpClient.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/SystemPropertyCsmConfigurationProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/Marshaller.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/MetricFilterInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/package-info.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/internal/RetryModeResolver.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/MapEntry.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/MonitoringEvent.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/Signer.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/SdkPlainSocketFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterHandler.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-



jar/com/amazonaws/auth/policy/Policy.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/Request.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/internal/JsonPolicyWriter.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/apache/request/impl/HttpGetWithBody.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/SimpleTypeStaxUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/cache/EndpointDiscoveryCacheLoader.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/VoidStaxUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/internal/JsonDocumentFields.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/TimestampFormat.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/conditions/StringCondition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/JsonUnmarshallerContext.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/AbstractFileTlsKeyManagersProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/metrics/RequestMetricCollector.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/handlers/RequestHandler2Adaptor.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/protocol/json/StructuredJsonMarshaller.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/Waiter.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/PropertiesFileCredentialsProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/conditions/BooleanCondition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/metrics/ServiceMetricCollector.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/WaiterUnrecoverableException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/retry/RetryPolicy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/endpointdiscovery/SystemPropertyEndpointDiscoveryProvider.java  
\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/TlsKeyManagersProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/conditions/DateCondition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterTimedOutException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/NoOpSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/CredentialsRequestHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/Utils/HttpContextUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/Resource.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/conditions/IpAddressCondition.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/ApiCallMonitoringEvent.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/internal/JsonPolicyReader.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/ThroughputMetricType.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/SimpleServiceMetricType.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/actions/package-info.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/ApiCallAttemptMonitoringEvent.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/Principal.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/VoidJsonUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/EndpointDiscoveryProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/SimpleMetricType.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/PolicyReaderOptions.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/CRC32ChecksumCalculatingInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/AcceptorPathMatcher.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/Condition.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

```

jar/com/amazonaws/retry/PredefinedRetryPolicies.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/policy/conditions/ConditionFactory.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/event/ProgressListenerChain.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/AmazonServiceException.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/monitoring/DefaultCsmConfigurationProviderChain.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/StringUtils.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/monitoring/MonitoringListener.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/metrics/SimpleThroughputMetricType.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/cache/Cache.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/http/conn/ClientConnectionManagerFactory.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/Presigner.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/cache/CacheLoader.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/endpointdiscovery/DefaultEndpointDiscoveryProviderChain.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/endpointdiscovery/EndpointDiscoveryProviderChain.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/metrics/AwsSdkMetrics.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/http/protocol/SdkHttpRequestExecutor.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/EndpointDiscoveryConfig.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/HttpClientConfig.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/BasicAWSCredentials.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/event/ProgressEvent.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/AWSRequestMetricsFullSupport.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/transform/JsonUnmarshallerContextImpl.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/SignerParams.java

```

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/CsmConfigurationProviderChain.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/impl/client/HttpRequestNoRetryHandler.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/StaxUnmarshallerContext.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/DefaultRequest.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/arn/ArnResource.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/DefaultServiceEndpointBuilder.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/config/InternalConfigJsonHelper.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/conditions/ArnCondition.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/monitoring/CsmConfigurationProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/SdkFunction.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/RequestMetricType.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/SignerTypeAware.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterBuilder.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterImpl.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AnonymousAWSCredentials.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/AmazonWebServiceRequest.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/HandlerBeforeAttemptContext.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/AwsClientSideMonitoringMetrics.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/Constants.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/arn/ArnConverter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/CompositeAcceptor.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/Wrapped.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

jar/com/amazonaws/waiters/WaiterParameters.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/IdentityEndpointBuilder.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/AWSRequestMetrics.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/client/builder/AdvancedConfig.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/policy/Statement.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/metrics/MetricInputStreamEntity.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/SignerFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/AmazonWebServiceClient.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/profile/internal/BasicProfileConfigFileLoader.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/FileStoreTlsKeyManagersProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/retry/internal/AuthErrorRetryStrategy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/event/ProgressEventType.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/VersionInfoUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/ProxyAuthenticationMethod.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/ResponseMetadataCache.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/XpathUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/StaxResponseHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/metrics/ServiceMetricType.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/waiters/WaiterExecutionBuilder.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/ServiceEndpointBuilder.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/monitoring/ProfileCsmConfigurationProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/QueryStringSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/monitoring/StaticCsmConfigurationProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/arn/Arn.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/metrics/ByteThroughputProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/endpointdiscovery/EnvironmentVariableEndpointDiscoveryProvider.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/policy/conditions/package-info.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/internal/MaxAttemptsResolver.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/PollingStrategy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/NoneTlsKeyManagersProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/NamespaceRemovingInputStream.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/ProcessCredentialsProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/ProgressTracker.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/request/ProgressSupport.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/ProgressInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/BasicProfileConfigLoader.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/internal/SignerKey.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/ProfileCredentialsProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/securitytoken/RoleInfo.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/securitytoken/STSPProfileCredentialsServiceLoader.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/ProfilesConfigFileWriter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/SyncProgressListener.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/LengthCheckInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/securitytoken/ProfileCredentialsService.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/RequestProgressInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/internal/SignerConstants.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/internal/AWS4SignerRequestParams.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/DelegateSSLSocket.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/ProgressEventFilter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/package-info.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/UnreliableFilterInputStream.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/request/Progress.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/Releasable.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/FIFOCache.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/DeliveryMode.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/UnreliableTestConfig.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkSSLSocket.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/event/SDKProgressPublisher.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

```
jar/com/amazonaws/util/Platform.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/internal/securitytoken/STSProfileCredentialsServiceProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/internal/ProfileProcessCredentialsProvider.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/internal/Profile.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/event/ResponseProgressInputStream.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/BoundedLinkedHashMap.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/ProfilesConfigFile.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/internal/AWS4SignerUtils.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/SdkSocket.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/internal/AbstractProfilesConfigFileScanner.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* =====
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
* =====
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
*/
```



Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/BasicNameValuePair.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/NameValuePair.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/URLEncodedUtils.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2016. Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/client/impl/ConnectionManagerAwareHttpClient.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/log/InternalLogFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/IdempotentUtils.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/client/impl/ApacheConnectionFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/client/impl/ApacheHttpClientFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/client/impl/SdkHttpClient.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/ssl/ShouldClearSslSessionPredicate.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/IRequestHandler2.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/FakeIOException.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/CapacityManager.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/DateTimeJsonSerializer.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkStructuredJsonFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/CanHandleNullCredentials.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/package-info.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/SignableRequest.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/ssl/privileged/PrivilegedMasterSecretValidator.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/SdkHttpUtils.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/DelegatingDnsResolver.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/client/ClientExecutionAbortTrackerTaskImpl.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/log/InternalLog.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/UriResourcePathUtils.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/StackedRequestHandler.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/client/HttpClientFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/request/HttpRequestAbortTaskTrackerImpl.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/PredefinedClientConfigurations.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/DelegateSocket.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/JsonErrorUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/settings/HttpClientSettings.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/client/ClientExecutionAbortTrackerTask.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/http/JsonErrorCodeParser.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ReadLimitInfo.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkInternalMap.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/client/ConnectionFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/TimeoutThreadPoolBuilder.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/SystemDefaultDnsResolver.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/request/HttpRequestAbortTask.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/HandlerContextKey.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/http/JsonErrorMessageParser.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/log/CommonsLogFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/ValidationUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/exception/HttpRequestTimeoutException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/client/NoOpClientExecutionAbortTrackerTask.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Base16Lower.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/client/ClientExecutionAbortTask.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/timers/client/ClientExecutionAbortTaskImpl.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/log/CommonsLog.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

jar/com/amazonaws/util/NumberUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/SdkThreadLocals.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/request/HttpRequestAbortTaskImpl.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkSSLMetricsSocket.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/SDKGlobalTime.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/client/ClientExecutionTimer.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/log/JulLogFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/DnsResolver.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/request/HttpRequestTimer.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/log/JulLog.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/DelegateInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/conn/ssl/MasterSecretValidators.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkMetricsSocket.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkThreadLocalsRegistry.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/JsonErrorResponseHandler.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/client/ClientExecutionTimeoutException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/ComparableUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/request/NoOpHttpRequestAbortTaskTracker.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkIOUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/JavaVersionParser.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/timers/request/HttpRequestAbortTaskTracker.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/conn/SdkConnectionKeepAliveStrategy.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/SdkInternalList.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/internal/CredentialsEndpointRetryPolicy.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/ContainerCredentialsFetcher.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/BaseCredentialsFetcher.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/CredentialsEndpointProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/InstanceMetadataServiceResourceFetcher.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/TokenBucket.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/ExceptionUtils.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/ContainerCredentialsRetryPolicy.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/EndpointPrefixAwareSigner.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/RetryMode.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/ContainerCredentialsProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/EC2ResourceFetcher.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/InstanceMetadataServiceCredentialsFetcher.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/EC2ContainerCredentialsProviderWrapper.java
```

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/ConnectionUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/internal/CredentialsEndpointRetryParameters.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

/\*\*

\* Interface for providing AWS credentials. Implementations are free to use any

\* strategy for providing AWS credentials, such as simply providing static

\* credentials that don't change, or more complicated implementations, such as

\* integrating with existing key management systems.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWSCredentialsProvider.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/RequestSigner.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ImmutableRequest.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/SignerAsRequestSigner.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights

\* Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/auth/SignerProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/auth/SignerProviderContext.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/adapters/types/StringToInputStreamAdapter.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/MetadataCache.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/auth/DefaultSignerProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/NullResponseMetadataCache.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/adapters/types/TypeAdapter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/adapters/types/StringToByteBufferAdapter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/auth/NoOpSignerProvider.java

No license file was found, but licenses were detected in source scan.

/\*

\*  
\* Copyright (c) 2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkJsonProtocolFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkCborGenerator.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/StructuredJsonGenerator.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkStructuredPlainJsonFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/JsonResponseHandler.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):



```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/ImmutableMapParameter.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/Builder.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/SignerConfigJsonHelper.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/Protocol.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/SignerConfig.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/config/JsonIndex.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
/**
* Interface for providing AWS region information. Implementations are free to use any strategy for
* providing region information.
*/
```

```
Found in path(s):
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/regions/AwsRegionProvider.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2016-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not
* use this file except in compliance with the License. A copy of the License is
* located at
*
* http://aws.amazon.com/apache2.0
*
*
```

\* or in the "license" file accompanying this file. This file is distributed on  
\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/http/CompositeErrorCodeParser.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/http/ErrorCodeParser.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/http/IonErrorCodeParser.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2019-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/waiters/WaiterExecutorServiceFactory.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/NamedDefaultThreadFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012-2023 Amazon Technologies, Inc.

\*

\* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt

\* for applicable license terms and NOTICE.txt for applicable notices.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at:

\*

\* <http://aws.amazon.com/apache2.0>

\*  
\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES  
\* OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/Md5Utils.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2013-2023 Amazon Technologies, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at:

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES

\* OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and

\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/regions/Regions.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/regions/ServiceAbbreviations.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt

\* for applicable license terms and NOTICE.txt for applicable notices.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

- \* express or implied. See the License for the specific language governing
- \* permissions and limitations under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Classes.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/DateUtils.java
- \*
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/BinaryUtils.java

No license file was found, but licenses were detected in source scan.

/\*

- \* Copyright 2015-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
- \*
- \* Licensed under the Apache License, Version 2.0 (the "License").
- \* You may not use this file except in compliance with the License.
- \* A copy of the License is located at
- \*
- \* <http://aws.amazon.com/apache2.0>
- \*
- \* or in the "license" file accompanying this file. This file is
- \* distributed
- \* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
- \* KIND, either
- \* express or implied. See the License for the specific language
- \* governing
- \* permissions and limitations under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/CredentialUtils.java
- No license file was found, but licenses were detected in source scan.

/\*

- \* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
- \*
- \* Licensed under the Apache License, Version 2.0 (the "License").
- \* You may not use this file except in compliance with the License.
- \* A copy of the License is located at
- \*
- \* <http://aws.amazon.com/apache2.0>
- \*
- \* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/Unmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/XmlUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/AWSCredentials.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/http/HttpMethodName.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/SigningAlgorithm.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/AmazonClientException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/auth/SignatureVersion.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/LegacyErrorUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/VoidUnmarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/transform/SimpleTypeUnmarshallers.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2019-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at:  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES  
\* OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/retry/ClockSkewAdjuster.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

jar/com/amazonaws/internal/SdkRequestRetryHeaderProvider.java  
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2012-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/CustomBackoffStrategy.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/ClasspathPropertiesFileCredentialsProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/http/IdleConnectionReaper.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/HandlerContextAware.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/InstanceProfileCredentialsProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/CRC32MismatchException.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/AWSCredentialsProviderChain.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/EC2MetadataClient.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/DefaultAWSCredentialsProviderChain.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/SystemPropertiesCredentialsProvider.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/DynamoDBBackoffStrategy.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/EnvironmentVariableCredentialsProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/AwsHostNameUtils.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/HostnameValidator.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/log/InternalLogApi.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
```

- \* You may not use this file except in compliance with the License.
- \* A copy of the License is located at
- \*
- \* <http://aws.amazon.com/apache2.0>
- \*
- \* or in the "license" file accompanying this file. This file is distributed
- \* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
- \* express or implied. See the License for the specific language governing
- \* permissions and limitations under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/ProfileStaticCredentialsProvider.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/DefaultMarshallingType.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/DefaultValueSupplier.java
- \*
- /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonOperationMetadata.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/OperationInfo.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/StringInputStream.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/builder/AwsAsyncClientBuilder.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/profile/path/config/SharedConfigDefaultLocationProvider.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/AwsRegionProviderChain.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/AwsSyncClientParams.java
- \*
- /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/MarshallingInfo.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/builder/AwsClientBuilder.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/RetryPolicyAdapter.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/v2/SimpleRetryPolicy.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/MapUnmarshaller.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/ClientHandler.java
- \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/AwsErrorResponseHandler.java



\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonErrorResponseMetadata.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/SdkClock.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonErrorShapeMetadata.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/internal/JsonMarshallerContext.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/PredefinedBackoffStrategies.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/builder/AwsSyncClientBuilder.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/AbstractErrorUnmarshaller.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/profile/path/AwsProfileFileLocationProviderChain.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/internal/SimpleTypePathMarshallers.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonContentTypeResolver.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/SdkBaseException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/StaticSignerProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/TimingInfoUnmodifiable.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/DefaultAwsRegionProviderChain.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/Protocol.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/profile/path/AwsProfileFileLocationProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/profile/path/cred/CredentialsLegacyConfigLocationProvider.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/internal/EmptyBodyJsonMarshaller.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/ProfileKeyConstants.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/MarshallLocation.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/presign/PresignerFacade.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/request/HttpRequestFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/SdkJsonGenerator.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/ProtocolMarshaller.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/builder/ExecutorFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/JsonContentTypeResolverImpl.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/StaticCredentialsProvider.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/response/AwsResponseHandlerAdapter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ClientConfigurationFactory.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWSRefreshableSessionCredentials.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/profile/path/cred/CredentialsEnvVarOverrideLocationProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/json/internal/SimpleTypeJsonMarshallers.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/AwsAsyncClientParams.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/v2/RetryOnStatusCodeCondition.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/v2/FixedDelayBackoffStrategy.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/profile/internal/AllProfiles.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWSStaticCredentialsProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/client/ClientExecutionParams.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/V2CompatibleBackoffStrategyAdapter.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/AwsEnvVarOverrideRegionProvider.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/StructuredPojo.java

\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/retry/v2/RetryCondition.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/transform/PathMarshallers.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/protocol/ProtocolRequestMarshaller.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-

jar/com/amazonaws/client/ClientHandlerImpl.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/transform/ListUnmarshaller.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/RequestConfig.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/SdkClientException.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/protocol/json/SdkJsonMarshallerFactory.java  
 \*  
 /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/retry/v2/RetryOnExceptionsCondition.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/retry/v2/AndRetryCondition.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/profile/path/cred/CredentialsDefaultLocationProvider.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/client/ClientHandlerParams.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/protocol/json/SdkStructuredJsonFactoryImpl.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/auth/profile/internal/BasicProfile.java  
 \*  
 /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/protocol/json/JsonClientMetadata.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/protocol/json/internal/MarshallerRegistry.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/http/ExecutionContext.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/retry/v2/BackoffStrategy.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/util/TimingInfo.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/internal/SdkFunction.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/auth/profile/internal/AwsProfileNameLoader.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/util/TimingInfoFullSupport.java  
 \*  
 /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/AmazonWebServiceResult.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/protocol/json/internal/Headermarshallers.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/retry/v2/RetryPolicyContext.java  
 \* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
 jar/com/amazonaws/regions/AwsSystemPropertyRegionProvider.java

```

* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/v2/OrRetryCondition.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/AmazonWebServiceRequestAdapter.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/util/XMLWriter.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/MarshallingType.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/presign/PresignerParams.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/http/timers/client/SdkInterruptedException.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/JsonContent.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/V2CompatibleBackoffStrategy.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/internal/ValueToStringConverters.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/profile/path/AwsDirectoryBasePathProvider.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/v2/MaxNumberOfRetriesCondition.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/regions/EndpointToRegion.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/http/SdkHttpMetadata.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/JsonProtocolMarshallerBuilder.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/auth/profile/internal/ProfileAssumeRoleCredentialsProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/retry/v2/RetryPolicy.java
*
/opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/internal/QueryParamMarshallers.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/regions/InstanceMetadataRegionProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/profile/path/config/ConfigEnvVarOverrideLocationProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/regions/AwsProfileRegionProvider.java
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/internal/JsonMarshaller.java
No license file was found, but licenses were detected in source scan.

```

```
/*
```

\* Copyright 2014-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt  
\* for applicable license terms and NOTICE.txt for applicable notices.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/ResettableInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/internal/ReleasableInputStream.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2013-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/regions/RegionMetadata.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/Base32.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-  
jar/com/amazonaws/util/Base16.java  
\*

/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/EncodingSchemeEnum.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/RegionAwareSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/EC2MetadataUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/SdkRuntime.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/MetricAware.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Throwables.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/IOUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Base16Codec.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Base64.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/CodecUtils.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/AbortedException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/jmx/SdkMBeanRegistrySupport.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Base32Codec.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkDigestInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/jmx/MBeans.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/Region.java  
\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/ResetException.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkBufferedInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/EncodingScheme.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/RegionMetadataParser.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkFilterOutputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/AbstractBase32Codec.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/regions/RegionUtils.java

\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/MetricsInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/SdkFilterInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/HttpClientWrappingInputStream.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/AWS4Signer.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Base64Codec.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/auth/ServiceAwareSigner.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/internal/ListWithAutoConstructFlag.java

\*  
/opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/Codec.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2014-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/JodaTime.java  
\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/StringMapBuilder.java

No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright 2018-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at

\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/util/PolicyUtils.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2016-2019. Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/Utils/Utils.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/apache/request/impl/HttpRequestFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014-2023 Amazon Technologies, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at:

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES

\* OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and



\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/ssl/TLSProtocol.java

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/http/conn/ssl/SdkTLSSocketFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/handlers/AsyncHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2015-2023 Amazon Technologies, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at:

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES

\* OR CONDITIONS OF ANY KIND, either express or implied. See the

\* License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677332\_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-jar/com/amazonaws/annotation/package-info.java

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/internal/SdkSSLContext.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License").
```

```
* You may not use this file except in compliance with the License.
```

```
* A copy of the License is located at
```

```
*
```

```
* http://aws.amazon.com/apache2.0
```

```
*
```

```
* or in the "license" file accompanying this file. This file is disalibuted
```

```
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
```

```
* express or implied. See the License for the specific language governing
```

```
* permissions and limitations under the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1620677332_1680116258.6110294/0/aws-java-sdk-core-1-12-430-sources-
jar/com/amazonaws/protocol/json/internal/JsonProtocolMarshaller.java
```

# 1.150 pgv-java-grpc 0.9.1

## 1.150.1 Available under license :

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

# 1.151 kotlin-reflect 1.8.10-release-430

## 1.151.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2010-2017 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaField.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/types/DescriptorSubstitutor.java
```

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/util/scopeUtils.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ExtensionReceiver.java

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/builtins/functions/BuiltInFictitiousFunctionClassFactory.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/FieldOverridabilityCondition.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/JavaDescriptorUtil.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/EmptyContainerForLocal.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ModuleDescriptor.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageViewDescriptor.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedAnnotations.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/ModuleClassResolver.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeErrorReporter.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/storage/ObservableStorageManager.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/KotlinJvmBinaryPackageSourceElement.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/SubstitutingScope.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/LazyScopeAdapter.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/RuntimeTypeMapper.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/SupertypeLoopChecker.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaCallableMemberDescriptor.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaMethodDescriptor.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationDescriptorImpl.java

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/CompositionTypeSubstitution.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/VariableAccessorDescriptor.kt





jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyDescriptor.java  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/Utils/SmartSet.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/Utils/CoreLib.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/KClassImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/LazyPackageViewDescriptorImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/KCallablesJvm.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/load/kotlin/header/ReadKotlinClassHeaderAnnotationVisitor.java  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaResolverCache.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/TypeParameterDescriptorImpl.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/types/TypeConstructor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/KParameterImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaPackageFragment.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/ParameterDescriptor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/resolve/constants/ConstantValueFactory.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/LocalClassifierTypeSettings.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/builtins/functions/FunctionInvokeDescriptor.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorVisitor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ImplicitReceiver.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/reflectLambda.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/impl/descriptors/VariableDescriptorWithAccessors.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-  
jar/kotlin/reflect/jvm/internal/KPackageImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```

jar/kotlin/reflect/jvm/internal/impl/descriptors/ConstUtil.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/DescriptorVisibility.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationAndConstantLoaderImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/components/DescriptorResolverUtils.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaPropertyDescriptor.java
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/types/DisjointKeysUnionTypeSubstitution.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/SubpackagesScope.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ValueParameterDescriptorImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/CompositePackageFragmentProvider.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentProvider.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ClassDescriptorBase.java
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorWithVisibility.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/constants/constantValues.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckerProcedureCallbacksImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/JavaDescriptorVisibilities.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/moduleByClassLoader.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationWithTarget.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/SimpleFunctionDescriptorImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/ReflectionObjectRenderer.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/JavaIncompatibilityRulesOverridabilityCondition.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/utils/WrappedValues.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentDescriptor.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/CallableMemberDescriptor.java

```

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ReceiverParameterDescriptor.java  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/Utils/Printer.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/TypeParameterDescriptor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ValueParameterDescriptor.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaElement.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ErrorReporter.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PackageFragmentDescriptorImpl.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorVisitorEmptyBodies.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/ReflectJvmMapping.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassOrPackageFragmentDescriptor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationAndConstantLoader.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ReceiverValue.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedPackageMemberScope.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/storage/StorageManager.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/MemberComparator.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/storage/storage.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationArgumentVisitor.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/typeParameterUtils.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyAccessorDescriptorImpl.java  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/DeserializedPackageFragment.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/util.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/ResolutionScope.kt

```

* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedMemberScope.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/resolvers.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/VisibilityUtil.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/KDeclarationContainerImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractReceiverParameterDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/MemberScopeImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/CastImplicitClassReceiver.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentProviderImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaAnnotationMapper.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/ChainedMemberScope.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/Receiver.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/VariableDescriptorImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/kotlin/DeserializedDescriptorResolver.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/KFunctionImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorNonRootImpl.java
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckingProcedureCallbacks.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/builtins/DefaultBuiltIns.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/DeclaredMemberIndex.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/util/ModuleVisibilityHelper.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/KClassesJvm.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/util.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```

```

jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractLazyTypeParameterDescriptor.java
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/KProperty2Impl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotatedImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/MemberDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyGetterDescriptorImpl.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/jvm/JavaDescriptorResolver.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyAccessorDescriptor.java
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/AbstractReceiverValue.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorNonRoot.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ModuleDescriptorImpl.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/DescriptorFactory.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/CallableDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/LazyJavaAnnotations.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/ExternalOverridabilityCondition.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/InnerClassesScopeWrapper.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/storage/NoLock.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/sources/JavaSourceElementFactory.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/ReflectProperties.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyGetterDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/KProperty0Impl.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/annotationUtil.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/LazyClassReceiverParameterDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```



- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/DescriptorEquivalenceForOverrides.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/dynamicTypes.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/FlexibleTypeDeserializer.kt
- \*
- /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/AbstractTypeConstructor.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractTypeParameterDescriptor.java
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/TypeAliasDescriptor.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassifierDescriptor.java
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/typeEnhancement/typeEnhancement.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/OverridingStrategy.kt
- \*
- /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/RawType.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeProjectionBase.java
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeSubstitutor.java
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaPropertyInitializerEvaluator.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/constants/CompileTimeConstant.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/CapturedTypeApproximation.kt
- \*
- /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/renderer/DescriptorRendererOptionsImpl.kt
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/AbstractClassTypeConstructor.java
- \* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/SuperCallReceiverValue.kt



\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/context.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeSubstitution.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassConstructorDescriptor.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/Utils.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractTypeAliasDescriptor.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractClassDescriptor.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/context.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/types/JavaTypeResolver.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/DeserializationComponentsForJava.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/calls/inference/CapturedTypeConstructor.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/ErasedOverridabilityCondition.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaTypeParameterDescriptor.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeCapabilities.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationsImpl.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/Utils/functions.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/Renderer/ClassifierNamePolicy.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/full/KTypes.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaAnnotationDescriptor.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/KotlinTypeChecker.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/FunctionPlaceholders.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/Annotations.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedTypeParameterDescriptor.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/JvmPackageScope.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/flexibleTypes.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/KClassifierImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/IntersectionTypeConstructor.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckingProcedure.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/full/KClasses.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassifierDescriptorWithTypeParameters.java  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/EmptyPackageFragmentDescriptor.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/ClassicTypeCheckerState.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/MetadataPackageFragmentProvider.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/KTypeParameterImpl.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/GivenFunctionsMemberScope.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/methodSignatureBuildingUtils.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/KTypesJvm.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/TypeIntersectionScope.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/types/RawType.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AbstractDeserializedPackageFragmentProvider.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/impl/SmartList.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationDeserializer.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/KotlinReflectionInternalError.kt  
\*  
/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/full/exceptions.kt  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/KotlinTypeCheckerImpl.java  
\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```
jar/kotlin/reflect/jvm/internal/impl/Utils/DFS.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/resolve/constants/IntegerValueTypeConstructor.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/ValueDescriptor.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/kotlin/JavaFlexibleTypeDeserializer.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/full/KClassifiers.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
// A seemingly obvious way to come about this case would be to declare a special exception class, but the problem is
that
```

```
Found in path(s):
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/storage/LockBasedStorageManager.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/NewCapturedType.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/resolve/OverridingUtil.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/checker/NewKotlinTypeChecker.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaMember.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaMethod.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/SpecialTypes.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/InvalidModuleException.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectKotlinClass.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaClass.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/container/DefaultImplementation.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/typeEnhancement/signatureEnhancement.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaAnnotation.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeUtils.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaConstructor.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/types/TypeWithEnhancement.kt

\*

/opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DescriptorDerivedFromTypeAlias.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/Utils.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyDescriptorImpl.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectKotlinClassFinder.kt

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaClassConstructorDescriptor.java

\* /opt/cola/permits/1620677371\_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```

jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/reflectClassUtil.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaPackageScope.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ContractDeserializer.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaWildcardType.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaPrimitiveType.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/Substitutable.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaStaticClassScope.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/deserialization/PlatformDependentDeclarationFilter.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/TypeAliasConstructorDescriptor.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/DescriptorVisibilities.java
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaClassifierType.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaAnnotationArguments.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaValueParameter.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/types/checker/IntersectionType.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/util/modifierChecks.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaScope.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/ModalityUtils.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaTypeParameter.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/builtins/BuiltInsLoader.kt
*
/opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/load/java/AnnotationTypeQualifierResolver.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/builtins/BuiltInsPackageFragment.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-
jar/kotlin/reflect/jvm/internal/impl/descriptors/deserialization/AdditionalClassPartsProvider.kt
* /opt/cola/permits/1620677371_1680210407.2450173/0/kotlin-reflect-1-8-10-sources-

```



## 1.152.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>

Found in path(s):

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/META-INF/maven/com.amazonaws/jmespath-java/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathExpression.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathSubExpression.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/OpLessThan.java

\*

/opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathLengthFunction.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/OpEquals.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathLiteral.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathIdentity.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathContainsFunction.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/JmesPathMultiSelectList.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-jar/com/amazonaws/jmespath/Comparator.java

\* /opt/cola/permits/1620677336\_1680234980.383754/0/jmespath-java-1-12-430-sources-

```
jar/com/amazonaws/jmespath/JmesPathProjection.java
*
/opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/OpGreaterThan.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathNotExpression.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/NumericComparator.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathFlatten.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathFilter.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/OpGreaterThanOrEqualTo.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/CamelCaseUtils.java
*
/opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathAndExpression.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathField.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathFunction.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/OpNotEquals.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/OpLessThanOrEqualTo.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/InvalidTypeException.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathEvaluationVisitor.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathVisitor.java
*
/opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/JmesPathValueProjection.java
* /opt/cola/permits/1620677336_1680234980.383754/0/jmespath-java-1-12-430-sources-
jar/com/amazonaws/jmespath/ObjectMapperSingleton.java
```

## 1.153 aws-java-sdk-sts 1.12.430

### 1.153.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2014-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*



\* Licensed under the Apache License, Version 2.0 (the "License").  
\* You may not use this file except in compliance with the License.  
\* A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed  
\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express or implied. See the License for the specific language governing  
\* permissions and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/internal/STSPProfileCredentialsService.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2012-2023 Amazon Technologies, Inc.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at:  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES  
\* OR CONDITIONS OF ANY KIND, either express or implied. See the  
\* License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/auth/SessionCredentialsProviderFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2018-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with  
\* the License. A copy of the License is located at  
\*  
\* <http://aws.amazon.com/apache2.0>  
\*  
\* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT  
\* WARRANTIES OR

\* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions  
\* and limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/PackedPolicyTooLargeExceptionUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/IDPCommunicationErrorException.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/ExpiredTokenExceptionUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/InvalidAuthorizationMessageExceptionUnmarshaller.java  
va  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AbstractAWSSecurityTokenService.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/MalformedPolicyDocumentExceptionUnmarshaller.java  
a  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/MalformedPolicyDocumentException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithWebIdentityRequest.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceClientBuilder.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/DecodeAuthorizationMessageResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithWebIdentityResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/GetAccessKeyInfoRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/GetFederationTokenResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleRequestMarshaller.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/FederatedUserStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetSessionTokenRequestMarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsyncClientBuilder.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/DecodeAuthorizationMessageResultStaxUnmarshaller.java  
ava

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithWebIdentityRequestMarshaller.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/auth/policy/actions/SecurityTokenServiceActions.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/InvalidIdentityTokenExceptionUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/IDPRejectedClaimExceptionUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/PackedPolicyTooLargeException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/FederatedUser.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsyncClient.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/IDPRejectedClaimException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/GetSessionTokenRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithWebIdentityResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithSAMLResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/RegionDisabledException.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetFederationTokenResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/DecodeAuthorizationMessageRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/RegionDisabledExceptionUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/DecodeAuthorizationMessageRequestMarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/IDPCommunicationErrorExceptionUnmarshaller.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/PolicyDescriptorTypeStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetFederationTokenRequestMarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/AWSSecurityTokenServiceException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-

jar/com/amazonaws/services/securitytoken/model/GetAccessKeyInfoResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/PolicyDescriptorType.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityResult.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/CredentialsStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/ExpiredTokenException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/AssumeRoleRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithSAMLResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/GetCallerIdentityRequestMarshaller.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/Credentials.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/InvalidIdentityTokenException.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/GetAccessKeyInfoRequestMarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/GetAccessKeyInfoResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/package-info.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/InvalidAuthorizationMessageException.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/AssumedRoleUser.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/TagStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithSAMLRequest.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/Tag.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithSAMLRequestMarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/AssumedRoleUserStaxUnmarshaller.java  
\*  
/opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/GetCallerIdentityResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-

jar/com/amazonaws/services/securitytoken/model/GetSessionTokenResult.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/AbstractAWSSecurityTokenServiceAsync.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/transform/GetSessionTokenResultStaxUnmarshaller.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/model/AssumeRoleResult.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/auth/STSAssumeRoleWithWebIdentitySessionCredentialsProvider.java  
\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-  
jar/com/amazonaws/services/securitytoken/RegionalEndpointsOptionResolver.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2011-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-

```
jar/com/amazonaws/auth/ShouldDoBlockingSessionRefresh.java
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/ShouldDoAsyncSessionRefresh.java
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/DaemonThreadFactory.java
*
/opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/RefreshableTask.java
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/SessionCredentialsHolder.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2018-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with
* the License. A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/
/**
* <p>
* An IAM policy in JSON format that you want to use as an inline session policy.
* </p>
* <p>
* You must pass an inline or managed <a
* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#policies_session">session
policy
* to this operation. You can pass a single JSON policy
document to use as an inline session policy. You can also
* specify up to 10 managed policy Amazon Resource Names (ARNs) to use as managed session policies.
* </p>
* <p>
* This parameter is optional. However, if you do not pass any session policies, then the resulting federated user
* session has no permissions.
* </p>
* <p>
* When you pass session policies, the session permissions are the intersection of the IAM user policies and the
* session policies that you pass. This gives you a way to further restrict the permissions for a federated user.
* You cannot use session policies to grant more permissions than those that are defined in the permissions policy
* of the IAM user. For more information, see <a
```

\* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access\_policies.html#policies\_session">Session Policies</a> in the <i>IAM User Guide</i>.

</p>

<p>

\* The resulting credentials can be used to access a resource that has a resource-based policy. If that policy

\* specifically references the federated user session in the <code>Principal</code> element of the policy, the session has the permissions allowed by the policy. These permissions are granted in addition to the permissions that are granted by the session policies.

</p>

<p>

\* The plaintext that you use for both inline and managed session policies can't exceed 2,048 characters. The JSON policy characters can be any ASCII character from the space character to the end of the valid character list ( through \u00FF). It can also include the tab ( ), linefeed ( ), and carriage return ( ) characters.

</p>

<note>

<p>

\* An Amazon Web Services conversion compresses the passed inline session policy, managed policy ARNs, and session tags into a packed binary format that has a separate limit. Your request can fail for this limit even if your plaintext meets the other requirements. The <code>PackedPolicySize</code> response element indicates by percentage how close the policies and tags for your request are to the upper size limit.

</p>

</note>

\*/

Found in path(s):

\* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/model/GetFederationTokenRequest.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2018-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with

\* the License. A copy of the License is located at

\*

\* http://aws.amazon.com/apache2.0

\*

\* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR

\* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions

\* and limitations under the License.

\*/

/\*\*

\* <p>

\* Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. A typical use is in a proxy application that gets temporary security credentials on behalf of distributed applications inside a corporate network. You must call the

\* `GetFederationToken` operation using the long-term security credentials of an IAM user. As a result,

\* this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based application. For a comparison of `GetFederationToken` with the other API operations that produce temporary credentials, see [Requesting Temporary Security Credentials](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html) and [Comparing the Amazon Web Services STS API operations](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#stsapi_comparison) in the *IAM User Guide*.

\* </p>

\* <note>

\* <p>

\* You can create a mobile-based or browser-based app that can authenticate users using a web identity provider like Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider. In this case, we recommend that you use [Amazon Cognito](http://aws.amazon.com/cognito/) or `AssumeRoleWithWebIdentity`. For more information, see [Federation Through a Web-based Identity Provider](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_assumerolewithwebidentity) in the *IAM User Guide*.

\* </p>

\* </note>

\* <p>

\* You can also call `GetFederationToken` using the security credentials of an Amazon Web Services account root user, but we do not recommend it. Instead, we recommend that you create an IAM user for the purpose of the proxy application. Then attach a policy to the IAM user that limits federated users to only the actions and resources that they need to access. For more information, see [IAM Best Practices](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html) in the *IAM User Guide*.

\* </p>

\* <p>

\* <b>Session duration</b>

\* </p>

\* <p>



\* The temporary credentials are valid for the specified duration, from 900 seconds (15 minutes) up to a maximum of 129,600 seconds (36 hours). The default session duration is 43,200 seconds (12 hours). Temporary credentials obtained by using the Amazon Web Services account root user credentials have a maximum duration of 3,600 seconds (1 hour).

</p>

<p>

<b>Permissions</b>

</p>

<p>

\* You can use the temporary credentials created by <code>GetFederationToken</code> in any Amazon Web Services service with the following exceptions:

</p>

<ul>

<li>

<p>

\* You cannot call any IAM operations using the CLI or the Amazon Web Services API. This limitation does not apply to console sessions.

</p>

</li>

<li>

<p>

\* You cannot call any STS operations except <code>GetCallerIdentity</code>.

</p>

</li>

</ul>

<p>

\* You can use temporary credentials for single sign-on (SSO) to the console.

</p>

<p>

\* You must pass an inline or managed <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access\_policies.html#policies\_session">session policy</a> to this operation. You can pass a single JSON policy document to use as an inline session policy. You can also specify up to 10 managed policy Amazon Resource Names (ARNs) to use as managed session policies. The plaintext that you use for both inline and managed session policies can't exceed 2,048 characters.

</p>

<p>

\* Though the session policy parameters are optional, if you do not pass a policy, then the resulting federated user session has no permissions. When you pass session policies, the session permissions are the intersection of the IAM user policies and the session policies that you pass.

This gives you a way to further restrict the permissions for a federated user. You cannot use session policies to grant more permissions than those that are defined in the permissions policy of the IAM user. For more information, see <a

\* [https://docs.aws.amazon.com/IAM/latest/UserGuide/access\\_policies.html#policies\\_session](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#policies_session)>Session Policies</a> in the <i>IAM User Guide</i>. For information about using <code>GetFederationToken</code> to create

- \* temporary security credentials, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_getfederationtoken">https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_getfederationtoken</a>

</p>

<p>

\* <b>GetFederationToken—Federation Through a Custom Identity Broker</b></a>.

</p>

<p>

\* You can use the credentials to access a resource that has a resource-based policy. If that policy specifically references the federated user session in the <code>Principal</code> element of the policy, the session has the permissions

allowed by the policy. These permissions are granted in addition to the permissions granted by the

- \* session policies.

</p>

<p>

\* <b>Tags</b></p>

</p>

<p>

\* (Optional) You can pass tag key-value pairs to your session. These are called session tags. For more information about session tags, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_session-tags.html">https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_session-tags.html</a>>Passing

- \* Session Tags in STS</a> in the <i>IAM User Guide</i>.

</p>

<note>

<p>

\* You can create a mobile-based or browser-based app that can authenticate users using a web identity provider like

- \* Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider. In this case, we recommend that you use <a href="http://aws.amazon.com/cognito/">http://aws.amazon.com/cognito/</a> or <code>AssumeRoleWithWebIdentity</code>. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_assumerolewithwebidentity">https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_assumerolewithwebidentity</a>

</p>

</note>

<p>

\* An administrator must grant you the permissions necessary to pass session tags. The administrator can also create

- \* granular permissions to allow you to pass only specific session tags. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial\_attribute-based-access-control.html">https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial\_attribute-based-access-control.html</a>>Tutorial:

- \* Using Tags for Attribute-Based Access Control</a> in the <i>IAM User Guide</i>.

</p>

<p>

\* Tag key–value pairs are not case sensitive, but case is preserved. This means that you cannot have separate

\* `Department` and `department` tag keys. Assume that the user that you are federating has

- \* the `Department=Marketing` tag and you pass the `department=engineering`

session tag. `Department` and `department` are not saved as separate tags, and the session tag passed in the request takes precedence over the user tag.

</p>

- \* `@param` getFederationTokenRequest
- \* `@return` Result of the GetFederationToken operation returned by the service.
- \* `@throws` MalformedPolicyDocumentException
  - \* The request was rejected because the policy document was malformed. The error message describes the specific error.
- \* `@throws` PackedPolicyTooLargeException
  - \* The request was rejected because the total packed size of the session policies and session tags combined was too large. An Amazon Web Services conversion compresses the session policy document, session policy ARNs, and session tags into a packed binary format that has a separate limit. The error message indicates by percentage how close the policies and tags are to the upper size limit. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_session-tags.html">Passing Session Tags in STS</a> in the <i>IAM User Guide</i>.</p>

<p>You could receive this error even though you meet other defined session policy and session tag limits. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\_iam-quotas.html#reference\_iam-limits-entity-length">IAM and STS Entity Character Limits</a> in the <i>IAM User Guide</i>.

- \* `@throws` RegionDisabledException
  - \* STS is not activated in the requested region for the account that is being asked to generate credentials. The account administrator must use the IAM console to activate STS in that region. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_enable-regions.html">Activating and Deactivating Amazon Web Services STS in an Amazon Web Services Region</a> in the <i>IAM User Guide</i>.
- \* `@sample` AWSSecurityTokenService.GetFederationToken
- \* `@see` <a href="http://docs.aws.amazon.com/goto/WebAPI/sts-2011-06-15/GetFederationToken" target="\_top">AWS API Documentation</a>

Found in path(s):

- \* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceClient.java
- \* /opt/cola/permits/1620677385\_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenService.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013-2023 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/policy/STSActions.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011-2023 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/WebIdentityFederationSessionCredentialsProvider.java
```

```
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/STSSessionCredentialsProvider.java
```

```
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/STSAssumeRoleSessionCredentialsProvider.java
```

```
*
```

```
/opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/auth/NoSessionSupportCredentials.java
```

```
* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
```

jar/com/amazonaws/auth/STSSessionCredentials.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2018-2023 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with

\* the License. A copy of the License is located at

\*

\* <http://aws.amazon.com/apache2.0>

\*

\* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR

\* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions

\* and limitations under the License.

\*/

/\*\*

\* <p>

\* Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. A typical use is in a proxy application that gets temporary security credentials on behalf of distributed applications inside a corporate network. You must call the

\* `GetFederationToken` operation using the long-term security credentials of an IAM user. As a result,

\* this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based

\* application. For a comparison of `GetFederationToken` with the other API operations that produce

\* temporary credentials, see <a

\* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html">Requesting Temporary

\* Security Credentials</a> and <a

\*

href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#stsapi\_comparison"

\* >Comparing the Amazon Web Services STS API operations</a> in the <i>IAM User Guide</i>.

\* </p>

\* <note>

\* <p>

\* You can create a mobile-based or browser-based app that can authenticate users using a web identity provider like

\* Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider.

In this case, we

\* recommend that you use <a href="http://aws.amazon.com/cognito/">Amazon Cognito</a> or

\* `AssumeRoleWithWebIdentity`. For more information, see <a href="

\*

https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_assumerolewithwebidentity"

\* >Federation Through a Web-based Identity Provider</a> in the <i>IAM User Guide</i>.

\* </p>

\* </note>

\* <p>

\* You can also call <code>GetFederationToken</code> using the security credentials of an Amazon Web Services account root user, but we do not recommend it. Instead, we recommend that you create an IAM user for the purpose of the proxy application. Then attach a policy to the IAM user that limits federated users to only the actions and resources that they need to access. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html">IAM Best Practices</a> in the <i>IAM User Guide</i>.

\* </p>

\* <p>

\* <b>Session duration</b>

\* </p>

\* <p>

\* The temporary credentials are valid for the specified duration, from 900 seconds (15 minutes) up to a maximum of 129,600 seconds (36 hours). The default session duration is 43,200 seconds (12 hours). Temporary credentials obtained by using the Amazon Web Services account root user credentials have a maximum duration of 3,600 seconds (1 hour).

\* </p>

\* <p>

\* <b>Permissions</b>

\* </p>

\* <p>

\* You can use the temporary credentials created by <code>GetFederationToken</code> in any Amazon Web Services service with the following exceptions:

\* </p>

\* <ul>

\* <li>

\* <p>

\* You cannot call any IAM operations using the CLI or the Amazon Web Services API. This limitation does not apply to console sessions.

\* </p>

\* </li>

\* <li>

\* <p>

\* You cannot call any STS operations except <code>GetCallerIdentity</code>.

\* </p>

\* </li>

\* </ul>

\* <p>

\* You can use temporary credentials for single sign-on (SSO) to the console.

\* </p>

\* <p>

\* You must pass an inline or managed <a

\* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access\_policies.html#policies\_session">session

policy</a>

\* to this operation. You can pass a single JSON policy document to use as an inline session policy. You can also

\* specify up to 10 managed policy Amazon Resource Names (ARNs) to use as managed session policies. The

plaintext

\* that you use for both inline and managed session policies can't exceed 2,048 characters.

\* </p>

\* <p>

\* Though the session policy parameters are optional, if you do not pass a policy, then the resulting federated user

\* session has no permissions. When you pass session policies, the session permissions are the intersection of the

\* IAM user policies and the session policies that you pass.

This gives you a way to further restrict the

\* permissions for a federated user. You cannot use session policies to grant more permissions than those that are

\* defined in the permissions policy of the IAM user. For more information, see <a

\* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access\_policies.html#policies\_session">Session

\* Policies</a> in the <i>IAM User Guide</i>. For information about using <code>GetFederationToken</code>

to create

\* temporary security credentials, see <a

\* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_temp\_request.html#api\_getfederationtoken"

>GetFederationToken—Federation Through a Custom Identity Broker</a>.

\* </p>

\* <p>

\* You can use the credentials to access a resource that has a resource-based policy. If that policy specifically

\* references the federated user session in the <code>Principal</code> element of the policy, the session has the

\* permissions

allowed by the policy. These permissions are granted in addition to the permissions granted by the

\* session policies.

\* </p>

\* <p>

\* <b>Tags</b>

\* </p>

\* <p>

\* (Optional) You can pass tag key-value pairs to your session. These are called session tags. For more information

\* about session tags, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_session-tags.html">Passing

\* Session Tags in STS</a> in the <i>IAM User Guide</i>.

\* </p>

\* <note>

\* <p>

\* You can create a mobile-based or browser-based app that can authenticate users using a web identity provider

like

```

* Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider. In this case, we
* recommend that you use Amazon Cognito or
* <code>AssumeRoleWithWebIdentity</code>. For more information, see <a href=
*
"https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_assumerolewithwebed
ntity"
* >Federation Through a Web-based Identity Provider in the <i>IAM User Guide</i>.
* </p>
* </note>
* <p>
* An administrator must grant you the permissions necessary to pass session tags. The administrator can also
create
* granular permissions to allow you to pass only specific session tags. For more information, see <a
* href="https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_attribute-based-access-
control.html">Tutorial:
* Using Tags for Attribute-Based Access Control in the <i>IAM User Guide</i>.
* </p>
* <p>
* Tag key–value pairs are not case sensitive, but case is preserved. This means that you cannot have separate
* <code>Department</code> and <code>department</code> tag keys. Assume that the user that you are
federating has
* the <code>Department</code>=<code>Marketing</code> tag and you pass the <code>department</code>=<
* <code>engineering</code>
session tag. <code>Department</code> and <code>department</code> are not saved as
* separate tags, and the session tag passed in the request takes precedence over the user tag.
* </p>
*
* @param getFederationTokenRequest
* @return A Java Future containing the result of the GetFederationToken operation returned by the service.
* @sample AWSSecurityTokenServiceAsync.GetFederationToken
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/sts-2011-06-15/GetFederationToken"
target="_top">AWS API
* Documentation
*/

```

Found in path(s):

```

* /opt/cola/permits/1620677385_1680116239.5470777/0/aws-java-sdk-sts-1-12-430-sources-
jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsync.java

```

## 1.154 opentelemetry-api 1.24.0

### 1.154.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright 2013-2020 The OpenZipkin Authors
*

```



\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677369\_1680116258.2154515/0/opentelemetry-api-1-24-0-sources-  
jar/io/opentelemetry/api/internal/ReadOnlyArrayMap.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2000-2021 JetBrains s.r.o.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <https://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1620677369\_1680116258.2154515/0/opentelemetry-api-1-24-0-sources-  
jar/io/opentelemetry/api/internal/Contract.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2008 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1620677369\_1680116258.2154515/0/opentelemetry-api-1-24-0-sources-jar/io/opentelemetry/api/internal/PercentEscaper.java

No license file was found, but licenses were detected in source scan.

// version.

Found in path(s):

\* /opt/cola/permits/1620677369\_1680116258.2154515/0/opentelemetry-api-1-24-0-sources-jar/io/opentelemetry/api/common/AttributesBuilder.java

## 1.155 re2j 1.7

### 1.155.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2021 The Go Authors. All rights reserved.

\*

\* Use of this source code is governed by a BSD-style

\* license that can be found in the LICENSE file.

\*/

Found in path(s):

\* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/MatcherInput.java

No license file was found, but licenses were detected in source scan.

<!--

Copyright (c) 2020 The Go Authors. All rights reserved.

Use of this source code is governed by a BSD-style

license that can be found in the LICENSE file.

-->

Found in path(s):

\* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/RE2J.gwt.xml

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2020 The Go Authors. All rights reserved.

- \* Use of this source code is governed by a BSD-style
- \* license that can be found in the LICENSE file.
- \*/

Found in path(s):

- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/UnicodeTables.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Prog.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Characters.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/RE2.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/MachineInput.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Unicode.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/CharGroup.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Parser.java
- \*
- /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-
- jar/com/google/re2j/super/com/google/re2j/Characters.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Simplify.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Utils.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/CharClass.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Regexp.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Machine.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Pattern.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Compiler.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Matcher.java
- \*
- /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-
- jar/com/google/re2j/PatternSyntaxException.java
- \* /opt/cola/permits/1620677354\_1680235716.3671062/0/re2j-1-7-sources-jar/com/google/re2j/Inst.java

# 1.156 kotlinpoet 1.7.2

## 1.156.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.157 xorg-randr 21.1.8-r0

## 1.157.1 Available under license :

The following is the 'standard copyright' agreed upon by most contributors, and is currently the canonical license preferred by the X.Org Foundation. This is a slight variant of the common MIT license form published by the Open Source Initiative at <http://www.opensource.org/licenses/mit-license.php>

Copyright holders of new code should use this license statement where possible, and insert their name to this list. Please sort by surname for people, and by the full name for other entities (e.g. Juliusz Chroboczek sorts before Intel Corporation sorts before Daniel Stone).

Copyright 2011 Dave Airlie  
Copyright 2000-2001 Juliusz Chroboczek  
Copyright 1998 Egbert Eich  
Copyright 2006-2007 Intel Corporation  
Copyright 2006 Nokia Corporation  
Copyright 2006-2008 Peter Hutterer  
Copyright 2006 Adam Jackson  
Copyright 2009-2010 NVIDIA Corporation  
Copyright 1987, 2003-2006, 2008-2010 Oracle and/or its affiliates.  
Copyright 1999 Keith Packard  
Copyright 2007-2009 Red Hat, Inc.  
Copyright 2005-2008  
Daniel Stone  
Copyright 2006-2009 Simon Thum  
Copyright 2003-2008, 2013 Geert Uytterhoeven  
Copyright 2006 Luc Verhaegen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.



THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following licenses are 'legacy': usually MIT/X11 licenses with the name of the copyright holder(s) in the license statement, but also some BSD-like licenses.

Copyright (C) 1994-2003 The XFree86 Project, Inc. All Rights Reserved.  
Copyright (C) Colin Harrison 2005-2008

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the XFree86 Project shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the XFree86 Project.

Copyright 1997 by The XFree86 Project, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that

copyright notice and this permission notice appear in supporting documentation, and that the name of the XFree86 Project, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The Xfree86 Project, Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE XFREE86 PROJECT, INC. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OREST ZBOROWSKI OR DAVID WEXELBLAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1985-1998, 2001 The Open Group  
Copyright 2002 Red Hat Inc., Durham, North Carolina.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

Copyright (c) 1987, 1989-1990, 1992-1995 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Copyright 2008 Tungsten Graphics, Inc., Cedar Park, Texas.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL TUNGSTEN GRAPHICS AND/OR ITS SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 1999-2000 SuSE, Inc.  
Copyright 2007 Red Hat, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of SuSE not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. SuSE makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

SUSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SUSE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1987-1991, 1993 by Digital Equipment Corporation, Maynard, Massachusetts.  
Copyright 1991 Massachusetts Institute of Technology, Cambridge, Massachusetts.  
Copyright 1991, 1993 Olivetti Research Limited, Cambridge, England.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts,  
Copyright 1994 Quarterdeck Office Systems.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Digital and Quarterdeck not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL AND QUARTERDECK DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1997 Digital Equipment Corporation.  
All rights reserved.

This software is furnished under license and may be used and copied only in accordance with the following terms and conditions. Subject to these conditions, you may download, copy, install, use, modify and distribute this software in source and/or binary form. No title or ownership is transferred hereby.

- 1) Any source code used, modified or distributed must reproduce and retain this copyright notice and list of conditions as they appear in the source file.
- 2) No right is granted to use any trade name, trademark, or logo of Digital Equipment Corporation. Neither the "Digital Equipment Corporation" name nor any trademark or logo of Digital Equipment Corporation may be used to endorse or promote products derived from this software without the prior written permission of Digital Equipment Corporation.
- 3) This software is provided "AS-IS" and any express or implied warranties, including but not limited to, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement are disclaimed. In no event shall DIGITAL be liable for any damages whatsoever, and in particular, DIGITAL shall not be liable for special, indirect, consequential, or incidental damages or damages for lost profits, loss of revenue or loss of use, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise, even

if advised of the possibility of such damage.

Copyright (c) 1991, 1996-1997 Digital Equipment Corporation, Maynard, Massachusetts.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY CLAIM, DAMAGES, INCLUDING, BUT NOT LIMITED TO CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Digital Equipment Corporation shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Digital Equipment Corporation.

SGI FREE SOFTWARE LICENSE B (Version 2.0, Sept. 18, 2008)  
Copyright (C) 1991-2000 Silicon Graphics, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software  
without restriction, including without limitation  
the rights to use, copy, modify, merge, publish, distribute, sublicense,  
and/or sell copies of the Software, and to permit persons to whom the  
Software is furnished to do so, subject to the following conditions:

The above copyright notice including the dates of first publication and either this permission notice or a reference to  
<http://oss.sgi.com/projects/FreeB/>  
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL

SILICON GRAPHICS, INC. BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1994, 1995 Hewlett-Packard Company

Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the Hewlett-Packard Company shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the Hewlett-Packard Company.

Copyright 1989 by Hewlett-Packard Company, Palo Alto, California.  
All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

HEWLETT-PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING

ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL HEWLETT-PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2001-2004 Red Hat Inc., Durham, North Carolina.

Copyright (c) 2003 by the XFree86 Project, Inc.

Copyright 2004-2005 Red Hat Inc., Raleigh, North Carolina.

All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL RED HAT AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2008 Red Hat, Inc.

Partly based on code Copyright 2000 SuSE, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Red Hat not be used in advertising or publicity pertaining to distribution of the



software without specific, written prior permission. Red Hat makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Red Hat DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Red Hat BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of SuSE not be used in advertising or publicity

pertaining to distribution of the software without specific, written prior permission. SuSE makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

SuSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SuSE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2006 Red Hat, Inc.

(C) Copyright 1998-1999 Precision Insight, Inc., Cedar Park, Texas.

All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation

the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL RED HAT, INC, OR PRECISION INSIGHT AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1995 X Consortium  
Copyright 2004 Red Hat Inc., Durham, North Carolina.  
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL RED HAT, THE X CONSORTIUM, AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Copyright 1998-2000 Precision Insight, Inc., Cedar Park, Texas.  
Copyright 2000 VA Linux Systems, Inc.  
Copyright (c) 2002, 2008, 2009 Apple Computer, Inc.  
Copyright (c) 2003-2004 Torrey T. Lyons.  
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a

copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL PRECISION INSIGHT AND/OR ITS SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(C) Copyright IBM Corporation 2003  
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL VA LINUX SYSTEM, IBM AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(C) Copyright IBM Corporation 2004-2005  
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL IBM, AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1997 Metro Link Incorporated

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the Metro Link shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Metro Link.

Copyright 1995-1998 by Metro Link, Inc.

Copyright (c) 1997 Matthieu Herrb

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Metro Link, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Metro Link, Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

METRO LINK, INC. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL METRO LINK, INC. BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1998 by Metro Link Incorporated

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Metro Link Incorporated not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Metro Link Incorporated makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

METRO LINK INCORPORATED DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL METRO LINK INCORPORATED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2000 by Conectiva S.A. (<http://www.conectiva.com>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE

SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL CONECTIVA LINUX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Conectiva Linux shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Conectiva Linux.

Copyright (c) 2001, Andy Ritger [aritger@nvidia.com](mailto:aritger@nvidia.com)  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of NVIDIA nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 1992 Vrije Universiteit, The Netherlands

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the Vrije Universiteit not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The Vrije Universiteit makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

The Vrije Universiteit DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT

SHALL The Vrije Universiteit BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1998 by Concurrent Computer Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Concurrent Computer Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Concurrent Computer Corporation makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

CONCURRENT COMPUTER CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CONCURRENT COMPUTER CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2004 Nokia

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Nokia not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Nokia makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

NOKIA DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NOKIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

(c)Copyright 1988,1991 Adobe Systems Incorporated.  
All rights reserved.

Permission to use, copy, modify, distribute, and sublicense this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notices appear in all copies and that both those copyright notices and this permission notice appear in supporting documentation and that the name of Adobe Systems Incorporated not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. No trademark license to use the Adobe trademarks is hereby granted. If the Adobe trademark "Display PostScript"(tm) is used to describe this software, its functionality or for any other purpose, such use shall be limited to a statement that this software works in conjunction with the Display PostScript system. Proper trademark attribution to reflect Adobe's ownership of the trademark shall be given whenever any such reference to the Display



PostScript system is made.

ADOBE MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE SOFTWARE FOR ANY PURPOSE. IT IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY. ADOBE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL ADOBE BE LIABLE TO YOU OR ANY OTHER PARTY FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, STRICT LIABILITY OR ANY OTHER ACTION ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. ADOBE WILL NOT PROVIDE ANY TRAINING OR OTHER SUPPORT FOR THE SOFTWARE.

Adobe, PostScript, and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.

Copyright 1989 Network Computing Devices, Inc., Mountain View, California.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of N.C.D. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. N.C.D. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (c) 1987 by the Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies. The University of California makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright 1992, 1993 Data General Corporation;  
Copyright 1992, 1993 OMRON Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the

above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that neither the name OMRON or DATA GENERAL be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission of the party whose name is to be used. Neither OMRON or DATA GENERAL make any representation about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OMRON AND DATA GENERAL EACH DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OMRON OR DATA GENERAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1998-2004, 2006 Keith Packard  
Copyright 2000-2002 Keith Packard, member of The XFree86 Project, Inc.  
Copyright (c) 2002 Apple Computer, Inc.  
Copyright (c) 2003 Torrey T. Lyons.  
All Rights Reserved.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1999 Keith Packard  
Copyright 2000 Compaq Computer Corporation

Copyright 2002 MontaVista Software Inc.

Copyright 2005 OpenedHand Ltd.

Copyright 2006 Nokia Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the authors and/or copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors and/or copyright holders make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHORS AND/OR COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO

THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHORS AND/OR COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1993 by Davor Matic

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Davor Matic makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (C) 2001-2004 Harold L Hunt II All Rights Reserved.

Copyright (C) Colin Harrison 2005-2008

Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL HAROLD L HUNT II BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Harold L Hunt II shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Harold L Hunt II.

Copyright 1990,91 by Thomas Roell, Dinkelscherben, Germany.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Thomas Roell not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Thomas Roell makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THOMAS ROELL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THOMAS ROELL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1990,91 by Thomas Roell, Dinkelscherben, Germany  
Copyright 1993 by David Wexelblat <dwex@goblin.org>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Thomas Roell and David Wexelblat

not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Thomas Roell and David Wexelblat makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THOMAS ROELL AND DAVID WEXELBLAT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THOMAS ROELL OR DAVID WEXELBLAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1990,91,92,93 by Thomas Roell, Germany.

Copyright 1991,92,93 by SGCS (Snitily Graphics Consulting Services), USA.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Thomas Roell nor SGCS be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Thomas Roell nor SGCS makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THOMAS ROELL AND SGCS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THOMAS ROELL OR SGCS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1998 by Alan Hourihane, Wigan, England.

Copyright 2000-2002 by Alan Hourihane, Flint Mountain, North Wales.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting

documentation, and that the name of Alan Hourihane not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Alan Hourihane makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

ALAN HOURIHANE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ALAN HOURIHANE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1995 Kaleb S. KEITHLEY

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL Kaleb S. KEITHLEY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Kaleb S. KEITHLEY shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Kaleb S. KEITHLEY

Copyright (c) 1997 Matthieu Herrb

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that

the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Matthieu Herrb not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Matthieu Herrb makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

MATTHIEU HERRB DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL MATTHIEU HERRB BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2004, Egbert Eich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL EGBERT EICH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Egbert Eich shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Egbert Eich.

Copyright 1993 by David Wexelblat <dwex@goblin.org>

Copyright 2005 by Kean Johnston <jkj@sco.com>

Copyright 1993 by David McCullough <davidm@stallion.oz.au>

Permission to use, copy, modify, distribute, and sell this software and its

documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of David Wexelblat not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. David Wexelblat makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

DAVID WEXELBLAT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DAVID WEXELBLAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1992 by Orest Zborowski <obz@Kodak.com>  
Copyright 1993 by David Wexelblat <dwex@goblin.org>

#### Permission

to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Orest Zborowski and David Wexelblat not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Orest Zborowski and David Wexelblat make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OREST ZBOROWSKI AND DAVID WEXELBLAT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OREST ZBOROWSKI OR DAVID WEXELBLAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1992 by Orest Zborowski <obz@Kodak.com>  
Copyright 1993 by David Dawes <dawes@xfree86.org>

Permission to use, copy, modify, distribute, and sell this software and its



documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Orest Zborowski and David Dawes not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Orest Zborowski and David Dawes make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OREST ZBOROWSKI AND DAVID DAWES DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS

SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OREST ZBOROWSKI OR DAVID DAWES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1995-1999 by Frederic Lepied, France. <fred@sugix.frmug.fr.net>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Frederic Lepied not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Frederic Lepied makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FREDERIC LEPIED DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FREDERIC LEPIED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1992 by Rich Murphey <Rich@Rice.edu>

Copyright 1993 by David Wexelblat <dwex@goblin.org>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that

copyright notice and this permission notice appear in supporting documentation, and that the names of Rich Murphey and David Wexelblat not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Rich Murphey and David Wexelblat make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

RICH MURPHEY AND DAVID WEXELBLAT DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL RICH MURPHEY OR DAVID WEXELBLAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1992 by Rich Murphey <Rich@Rice.edu>  
Copyright 1993 by David Dawes <dawes@xfree86.org>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Rich Murphey and David Dawes not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Rich Murphey and David Dawes make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

RICH MURPHEY AND DAVID DAWES DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL RICH MURPHEY OR DAVID DAWES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2003-2004 Anders Carlsson

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that

copyright notice and this permission notice appear in supporting documentation, and that the name of Anders Carlsson not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Anders Carlsson makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

ANDERS CARLSSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ANDERS CARLSSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 2003 Anders Carlsson  
Copyright 2003-2004 Eric Anholt  
Copyright 2004 Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Eric Anholt not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Eric Anholt makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

ERIC ANHOLT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ERIC ANHOLT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER DISCLAIMS ALL

WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2003-2004 Philip Blundell

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Philip Blundell not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Philip Blundell makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

PHILIP BLUNDELL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL PHILIP BLUNDELL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1994-2003 by The XFree86 Project, Inc.

Copyright 1997 by Metro Link, Inc.

Copyright 2003 by David H. Dawes.

Copyright 2003 by X-Oz Technologies.

Copyright (c) 2004, X.Org Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the

Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDER(S) OR AUTHOR(S) BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the copyright holder(s) and author(s) shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the copyright holder(s) and author(s).

Copyright 1990,91 by Thomas Roell, Dinkelscherben, Germany

Copyright 1993 by David Wexelblat <dwex@goblin.org>

Copyright 1999 by David Holland <davidh@iquest.net>

Copyright 2000 Compaq Computer Corporation

Copyright 2002 Hewlett-Packard Company

Copyright

2004, 2005 Red Hat, Inc.

Copyright 2004 Nicholas Miell

Copyright 2005 Trolltech AS

Copyright 2006 Intel Corporation

Copyright 2006-2007 Keith Packard

Copyright 2008 Red Hat, Inc

Copyright 2008 George Sapountzis <gsap7@yahoo.gr>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The copyright holders make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY

AND

FITNESS, IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS

SOFTWARE.

Copyright 2000 Keith Packard, member of The XFree86 Project, Inc.  
2005 Lars Knoll & Zack Rusin, Trolltech

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1987, 1998 The Open Group  
Copyright 1998-1999, 2001 The XFree86 Project, Inc.  
Copyright 2000 VA Linux Systems, Inc.  
Copyright (c) 2000, 2001 Nokia Home Communications  
Copyright 2007, 2008 Red Hat, Inc.  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT

OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Copyright 1996 by Thomas E. Dickey <dickey@clark.net>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the above listed copyright holder(s) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

THE ABOVE LISTED COPYRIGHT HOLDER(S) DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE ABOVE LISTED COPYRIGHT HOLDER(S) BE

LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1998-1999 Precision Insight, Inc., Cedar Park, Texas.

Copyright (c) 2001 Andreas Monitzer.

Copyright (c) 2001-2004 Greg Parker.

Copyright (c) 2001-2004 Torrey T. Lyons

Copyright (c) 2002-2003 Apple Computer, Inc.

Copyright (c) 2004-2005 Alexander Gottwald

Copyright (c) 2002-2009 Apple Inc.

Copyright (c) 2007 Jeremy Huddleston

All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a

copy of this software and associated documentation files (the "Software"),  
to deal in the Software without restriction, including without limitation  
the rights to use, copy, modify, merge, publish, distribute, sublicense,  
and/or sell copies of the Software, and to permit persons  
to whom the  
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
THE ABOVE LISTED COPYRIGHT HOLDER(S) BE LIABLE FOR ANY CLAIM, DAMAGES OR  
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,  
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER  
DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright  
holders shall not be used in advertising or otherwise to promote the sale,  
use or other dealings in this Software without prior written authorization.

Copyright (C) 1999,2000 by Eric Sunshine <sunshine@sunshineco.com>  
Copyright (C) 2001-2005 by Thomas  
Winischhofer, Vienna, Austria.

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products  
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN  
NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
SPECIAL,  
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED  
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR  
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF



LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2005 Bogdan D. bogdand@users.sourceforge.net

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the author shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the author.

Copyright 2002 David Dawes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the author(s) shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the author(s).

Copyright (C) 1996-1999 SciTech Software, Inc.

Copyright

(C) David Mosberger-Tang

Copyright (C) 1999 Egbert Eich

Copyright (C) 2008 Bart Trojanowski, Symbio Technologies, LLC

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the authors not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHORS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2005-2006 Luc Verhaegen.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDER(S) OR AUTHOR(S) BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 1995 by Robin Cutshaw <robin@XFree86.Org>

Copyright 2000 by Egbert Eich

Copyright 2002 by David Dawes

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of the above listed copyright holder(s) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The above listed copyright holder(s) make(s) no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE ABOVE LISTED COPYRIGHT HOLDER(S) DISCLAIM(S) ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE ABOVE LISTED COPYRIGHT HOLDER(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1997-2004 by Marc Aurele La France (TSI @ UQV), tsi@xfree86.org

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Marc Aurele La France not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Marc Aurele La France makes no representations about the suitability of this software for any purpose. It is provided "as-is" without express or implied warranty.

MARC AURELE LA FRANCE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,

INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL MARC AURELE LA FRANCE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1990, 1991 by Thomas Roell, Dinkelscherben, Germany

Copyright 1992 by David Dawes <dawes@XFree86.org>

Copyright 1992 by Jim Tsillas <jtsilla@damon.ccs.northeastern.edu>

Copyright 1992 by Rich Murphey <Rich@Rice.edu>

Copyright 1992 by Robert Baron <Robert.Baron@ernst.mach.cs.cmu.edu>

Copyright

1992 by Orest Zborowski <obz@eskimo.com>

Copyright 1993 by Vrije Universiteit, The Netherlands

Copyright 1993 by David Wexelblat <dwex@XFree86.org>

Copyright 1994, 1996 by Holger Veit <Holger.Veit@gmd.de>

Copyright 1997 by Takis Psarogiannakopoulos <takis@dpmms.cam.ac.uk>

Copyright 1994-2003 by The XFree86 Project, Inc

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of the above listed copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The above listed copyright holders make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE ABOVE

LISTED COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE ABOVE LISTED COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2001-2005 by J. Kean Johnston <jkj@sco.com>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting

documentation, and that the name J. Kean Johnston not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. J.

Kean Johnston makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

J. KEAN JOHNSTON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL J. KEAN JOHNSTON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 2000 Jakub Jelinek (jakub@redhat.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL JAKUB JELINEK BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 1997,1998 by UCHIYAMA Yasushi

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of UCHIYAMA Yasushi not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. UCHIYAMA Yasushi makes no representations about the suitability of this software for any purpose. It is provided

"as is" without express or implied warranty.

UCHIYAMA YASUSHI DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL UCHIYAMA YASUSHI BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 2000 Keith Packard

2004 Eric Anholt

2005 Zack Rusin

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Copyright holders make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

(C) Copyright IBM Corporation

2002-2007

All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next

paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that this permission notice appear in supporting documentation. This permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2007 OpenedHand Ltd

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of OpenedHand Ltd not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. OpenedHand Ltd makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OpenedHand Ltd DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OpenedHand Ltd BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1987, 1990, 1993

The

Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by  
Chris Torek.

This code is derived from software contributed to Berkeley by  
Michael Rendell of Memorial University of Newfoundland.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
4. Neither the name of the University nor the names of its contributors  
may be used to endorse or promote products derived from this software  
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS

``AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

## 1.158 bridgeutils 1.7.1-r2

### 1.158.1 Available under license :

/\*

\* Copyright (C) 2000 Lennert Buytenhek

\*

\* This program is free software; you can redistribute it and/or

\* modify it under the terms of the GNU General Public License as

\* published by the Free Software Foundation; either version 2 of the

\* License, or (at your option) any later version.

\*



- \* This program is distributed in the hope that it will be useful, but
- \* WITHOUT ANY WARRANTY; without even the implied warranty of
- \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
- \* General Public License for more details.
- \*
- \* You should have received a copy of the GNU General Public License
- \* along with this program; if not, write to the Free Software
- \* Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
- \*/

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.159 pax-utils 1.3.7-r1

### 1.159.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free



software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such

interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is

allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.
```

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# 1.160 jre 17.0.7+7-LTS

## 1.160.1 Available under license :

```
c-libutl 20160225
```

```
c-libutl License
```

```
...
```

This software is distributed under the terms of the BSD license.

```
== BSD LICENSE =====
```

(C) 2009 by Remo Dentato (rdentato@gmail.com)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<http://opensource.org/licenses/bsd-license.php>



...

## Apache Xerces v2.12.2

### Apache Xerces Notice

<pre>

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
=====
```

Apache Xerces Java  
Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:  
- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.  
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.  
- voluntary contributions made by Paul Eng on behalf of the  
Apache Software Foundation that were originally developed at iClick, Inc.,  
software copyright (c) 1999.

</pre>

### Apache 2.0 License

<pre>

```
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
```

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean  
an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but  
not limited to compiled object code, generated documentation,  
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or  
Object form, made available under the License, as indicated by a  
copyright notice that is included in or attached to the work  
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object  
form, that is based on (or derived from) the Work and for which the  
editorial revisions, annotations, elaborations, or other modifications  
represent, as a whole, an original work  
of authorship. For the purposes  
of this License, Derivative Works shall not include works that remain  
separable from, or merely link (or bind by name) to the interfaces of,  
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including  
the original version of the Work and any modifications or additions  
to that Work or Derivative Works thereof, that is intentionally  
submitted to Licensor for inclusion in the Work by the copyright owner  
or by an individual or Legal Entity authorized to submit on behalf of  
the copyright owner. For the purposes of this definition, "submitted"  
means any form of electronic, verbal, or written communication sent  
to the Licensor or its representatives, including but not limited to  
communication on electronic mailing lists, source code control systems,  
and issue tracking systems that are managed by, or on behalf of, the  
Licensor for the purpose of discussing and improving the Work,  
but  
excluding communication that is conspicuously marked or otherwise  
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity  
on behalf of whom a Contribution has been received by Licensor and  
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any

Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of

TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
</pre>
xwd v1.0.7
```

```
xwd utility
<pre>
```

This is the copyright for the files in src/java.desktop/unix/native/libawt\_xawt:  
list.h, multiVis.h, wsutils.h, list.c, multiVis.c

Copyright 1994 Hewlett-Packard Co.  
Copyright 1996, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

```
</pre>
IAIK (Institute for Applied Information Processing and Communication) PKCS#11 wrapper files v1
```

```
IAIK License
<pre>
```

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of Technology."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## The FreeType Project: Freetype v2.12.1

### FreeType Notice

...

FreeType comes with two licenses from which you can choose the one which fits your needs best.

The FreeType License (FTL) is the most commonly used one. It is a BSD-style license with a credit clause and thus compatible with the GNU Public License (GPL) version 3, but not with the

GPL version 2.

The GNU General Public License (GPL), version 2. Use it for all projects which use the GPLv2 also, or which need a license compatible to the GPLv2.

...

### FreeType License

...

Copyright (C) 1996-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2007-2022 by Dereg Clegg and Michael Toftdal.  
Copyright (C) 1996-2022 by Just van Rossum, David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2004-2022 by Masatake YAMATO and Redhat K.K.  
Copyright (C) 2007-2022 by Dereg Clegg and Michael Toftdal.  
Copyright (C) 2007-2022 by David Turner.  
Copyright (C) 2022 by David Turner,  
Robert Wilhelm, Werner Lemberg, and Moazin Khatti.  
Copyright (C) 2007-2022 by Rahul Bhalerao <rahul.bhalerao@redhat.com>, <b.rahul.pm@gmail.com>.  
Copyright (C) 2008-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and suzuki toshiya.  
Copyright (C) 2019-2022 by Nikhil Ramakrishnan, David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2009-2022 by Oran Agra and Mickey Gabel.  
Copyright (C) 2004-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and George Williams.  
Copyright (C) 2004-2022 by Masatake YAMATO, Red Hat K.K.,  
Copyright (C) 2003-2022 by Masatake YAMATO, Redhat K.K.,  
Copyright (C) 2013-2022 by Google, Inc.  
Copyright (C) 2018-2022 by David Turner, Robert Wilhelm, Dominik Rttisches, and Werner Lemberg.  
Copyright (C) 2005-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright 2013 by Google, Inc.

#### The FreeType Project LICENSE

-----

2006-Jan-27

Copyright 1996-2002, 2006 by  
David Turner, Robert Wilhelm, and Werner Lemberg

#### Introduction

=====

The FreeType Project is distributed in several archive packages;  
some of them may contain, in addition to the FreeType font engine,



various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. (as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. (royalty-free' usage)
- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
""
Portions of this software are copyright <year> The FreeType
Project (www.freetype.org). All rights reserved.
""
```

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms  
=====

0. Definitions

-----  
Throughout this license, the terms `package`, `FreeType Project`, and `FreeType archive` refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project`, be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

#### 1. No Warranty

-----  
THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

#### 2. Redistribution

-----  
This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

-----

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

### 4. Contacts

-----

There are two mailing lists related to FreeType:

- o freetype@nongnu.org

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution.

If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

o [freetype-devel@nongnu.org](mailto:freetype-devel@nongnu.org)

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

<http://www.freetype.org>

...

### GPL v2

...

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it.

By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to

surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally,  
any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program

is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply



in other  
circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

...

```
Additional Freetype Attributions
...
```

-----

The below license applies to the following files:  
libfreetype/src/psaux/psarrst.c

libfreetype/src/psaux/psarrst.h  
libfreetype/src/psaux/psblues.c  
libfreetype/src/psaux/psblues.h  
libfreetype/src/psaux/pserror.c  
libfreetype/src/psaux/pserror.h  
libfreetype/src/psaux/psfixed.h  
libfreetype/src/psaux/psfont.c  
libfreetype/src/psaux/psfont.h  
libfreetype/src/psaux/psft.c  
libfreetype/src/psaux/psft.h  
libfreetype/src/psaux/psglue.h  
libfreetype/src/psaux/pshints.c  
libfreetype/src/psaux/pshints.h  
libfreetype/src/psaux/psintrp.c  
libfreetype/src/psaux/psintrp.h  
libfreetype/src/psaux/psread.c  
libfreetype/src/psaux/psread.h  
libfreetype/src/psaux/psstack.c  
libfreetype/src/psaux/psstack.h  
libfreetype/src/psaux/pstypes.h

#### Copyright

2006-2014 Adobe Systems Incorporated.

This software, and all works of authorship, whether in source or object code form as indicated by the copyright notice(s) included herein (collectively, the "Work") is made available, and may only be used, modified, and distributed under the FreeType Project License, LICENSE.TXT. Additionally, subject to the terms and conditions of the FreeType Project License, each contributor to the Work hereby grants to any individual or legal entity exercising permissions granted by the FreeType Project License and this section (hereafter, "You" or "Your") a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such contributor that are necessarily infringed by their contribution(s) alone or by combination of their contribution(s) with the Work to which such contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

By using, modifying, or distributing the Work you indicate that you

have read and understood the terms and conditions of the  
FreeType Project License as well as those provided in this section,  
and you accept  
them fully.

...

### MIT License

...

-----  
The below license applies to the following files:  
libfreetype/include/freetype/internal/fthash.h  
libfreetype/src/base/fthash.c

Copyright 2000 Computing Research Labs, New Mexico State University  
Copyright 2001-2015

Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a  
copy of this software and associated documentation files (the "Software"),  
to deal in the Software without restriction, including without limitation  
the rights to use, copy, modify, merge, publish, distribute, sublicense,  
and/or sell copies of the Software, and to permit persons to whom the  
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY  
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT  
OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR  
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

...

## Harfbuzz v4.4.1

### Harfbuzz License

<https://github.com/harfbuzz/harfbuzz/blob/4.4.1/COPYING>

<pre>

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow.  
For parts of HarfBuzz that are licensed under different licenses see individual  
files names COPYING in subdirectories where applicable.

Copyright 2010-2022 Google, Inc.  
Copyright 2018-2020 Ebrahim Byagowi  
Copyright 2019-2020 Facebook, Inc.  
Copyright 2012-2015 Mozilla Foundation.  
Copyright 2011 Codethink Limited  
Copyright 2008-2010 Nokia Corporation and/or its subsidiary(-ies)  
Copyright 2009 Keith Stribley  
Copyright 2009 Martin Hosken and SIL International  
Copyright 2007 Chris Wilson  
Copyright 2005-2022 Behdad Esfahbod  
Copyright 2005 David Turner  
Copyright 2004-2013 Red Hat, Inc.  
Copyright 1998-2004 David Turner and Werner Lemberg  
Copyright 2016 Elie Roux <elie.roux@telecom-bretagne.eu>  
Copyright 2018-2019 Adobe Inc.  
Copyright 2018 Khaled Hosny  
Copyright 2016 Igalia S.L.

For  
full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without  
license or royalty fees, to use, copy, modify, and distribute this  
software and its documentation for any purpose, provided that the  
above copyright notice and the following two paragraphs appear in  
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR  
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES  
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN  
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH  
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,  
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND  
FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS  
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO  
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

All source

code, except for one section, is licensed as above. The one exception is licensed with a slightly different MIT variant:  
The contents of this directory are licensed under the following terms:

-----  
The below license applies to the following files:

libharfbuzz/hb-ucd.cc

Copyright (C) 2012 Grigori Goronzy <greg@kinoho.net>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

</pre>

## PC/SC Lite v1.9.5

### PC/SC Lite License

<pre>

The main parts of the code are using the BSD-like licence bellow:

Copyright (c) 1999-2003 David Corcoran <corcoran@linuxnet.com>

Copyright (c) 2001-2011 Ludovic Rousseau <ludovic.rousseau@free.fr>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Changes to this license can be made only by the copyright author with explicit written consent.

THIS

SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some source code files are using other licences. See the [COPYING](<https://salsa.debian.org/rousseau/PCSC/-/blob/master/COPYING>) file for details.

</pre>

```
The Unicode Standard, Unicode Character Database, Version 13.0.0
```

```
Unicode Character Database
```

```
...
```

## UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

## COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2020 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or



(b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

...

## Eastman Kodak Company: Portions of color management and imaging software

### Eastman Kodak Notice

<pre>

Portions Copyright Eastman Kodak Company 1991-2003

</pre>

## Thai Dictionary

### Thai Dictionary License

<pre>

Copyright (C) 1982 The Royal Institute, Thai Royal Government.

Copyright (C) 1998 National Electronics and Computer Technology Center,  
National Science and Technology Development Agency,  
Ministry of Science Technology and Environment,  
Thai Royal Government.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

## Apache Santuario v2.3.0

### Apache Santuario Notice

<pre>

Apache Santuario - XML Security for Java  
Copyright 1999-2021 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

It was originally based on software copyright (c) 2001, Institute for  
Data Communications Systems, <<http://www.nue.et-inf.uni-siegen.de/>>.

The development of this software was partly funded by the European  
Commission in the <WebSig> project in the ISIS Programme.

</pre>

### Apache 2.0 License

<pre>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity  
authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except  
in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

</pre>

#### ADDITIONAL INFORMATION ABOUT LICENSING

Certain files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception.

Note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0 and may include FreeType. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result.

Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Failing to distribute notices associated with some files may also create unexpected legal consequences.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

## DOM Level 3 Core Specification v1.0

### W3C Software Notice

<pre>

Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University).  
All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

</pre>

### W3C License

<pre>

## W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided

by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

- 1.The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- 2.Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- 3.Notice of any changes or modifications to the files, including the date changes were made.  
(We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS



MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS,COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

---

This formulation of W3C's notice and license became active on December 31 2002. This

version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to [site-policy@w3.org](mailto:site-policy@w3.org).

</pre>

## OASIS PKCS #11 Cryptographic Token Interface v3.0

### OASIS PKCS #11 Cryptographic Token Interface License

<pre>

Copyright OASIS Open 2020. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website: [<http://www.oasis-open.org/policies-guidelines/ipr>]

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright

notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an

attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

</pre>

## Cryptix AES v3.2.0

### Cryptix General License

<pre>

Cryptix General License

Copyright (c) 1995-2005 The Cryptix Foundation Limited.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

OPENJDK ASSEMBLY EXCEPTION

The OpenJDK source code made available by Oracle America, Inc. (Oracle) at [openjdk.java.net](http://openjdk.java.net) ("OpenJDK Code") is distributed under the terms of the GNU General Public License <<http://www.gnu.org/copyleft/gpl.html>> version 2 only ("GPL2"), with the following clarification and special exception.

Linking this OpenJDK Code statically or dynamically with other code is making a combined work based on this library. Thus, the terms and conditions of GPL2 cover the whole combination.

As a special exception, Oracle gives you permission to link this OpenJDK Code with certain code licensed by Oracle as indicated at <http://openjdk.java.net/legal/exception-modules-2007-05-08.html> ("Designated Exception Modules") to produce an executable, regardless of the license terms of the Designated Exception Modules, and to copy and distribute the resulting executable under GPL2, provided that the Designated Exception Modules continue to be governed by the licenses under which they were offered by Oracle.

As such, it allows licensees and sublicensees of Oracle's GPL2 OpenJDK Code to build an executable that includes those portions of necessary code that Oracle could not provide under GPL2 (or that Oracle has provided under GPL2 with the Classpath exception). If you modify or add to the OpenJDK code, that new GPL2 code may still be combined with Designated Exception Modules if the new code is made subject to this exception by its copyright holder.

## libpng v1.6.38

### libpng License

<pre>

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

=====

PNG Reference Library License version 2

-----

Copyright (c) 1995-2022 The PNG Reference Library Authors.

Copyright (c) 2018-2022 Cosmin Truta

Copyright (c) 1998-2018 Glenn Randers-Pehrson

Copyright (c) 1996-1997 Andreas Dilger

Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or

other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

-----  
libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Mans Rullgard  
Cosmin Truta  
Gilles Vollant  
James Yu  
Mandar Sahastrabudhe  
Google Inc.  
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000

Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing

Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

#### TRADEMARK

=====

The name "libpng" has not been registered by the Copyright owners as a trademark in any jurisdiction. However, because libpng has been distributed and maintained world-wide, continually since 1995, the Copyright owners claim "common-law trademark protection" in any jurisdiction where common-law trademark is recognized.

</pre>

#### ### AUTHORS File Information

^^^

#### PNG REFERENCE LIBRARY AUTHORS

=====

This is the list of PNG Reference Library ("libpng") Contributing Authors, for copyright and licensing purposes.

- \* Andreas Dilger
- \* Cosmin Truta
- \* Dave Martindale
- \* Eric S. Raymond
- \* Gilles Vollant

- \* Glenn Randers-Pehrson
- \* Greg Roelofs
- \* Guy Eric Schalnat
- \* James Yu
- \* John Bowler
- \* Kevin Bracey
- \* Magnus Holmgren
- \* Mandar Sahastrabuddhe
- \* Mans Rullgard
- \* Matt Sarett
- \* Mike Klein
- \* Pascal Massimino
- \* Paul Schmidt
- \* Qiang Zhou
- \* Sam Bushell
- \* Samuel Williams
- \* Simon-Pierre Cadieux
- \* Tim Wegner
- \* Tom Lane
- \* Tom Tanner
- \* Vadim Barkov
- \* Willem van Schaik
- \* Zhijie Liang
- \* Arm Holdings
  - Richard Townsend
- \* Google Inc.
  - Matt Sarett
  - Mike Klein
  - Dan Field
  - Sami Boukortt

The build projects, the build scripts, the test scripts, and other files in the "ci", "projects", "scripts" and "tests" directories, have other copyright owners, but are released under the libpng license.

Some files in the "contrib" directory, and some tools-generated files that are distributed with libpng, have other copyright owners, and are released under other open source licenses.

...

## Apache Xalan v2.7.2

### Apache Xalan Notice

<pre>

=====

=====

== NOTICE file corresponding to the section 4d of the Apache License, Version 2.0, ==

== in this case for the Apache Xalan distribution. ==



=====  
=====  
This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).

Specifically, we only include the XSLTC portion of the source from the Xalan distribution.  
The Xalan project has two processors: an interpretive one (Xalan Interpretive) and a  
compiled one (The XSLT Compiler (XSLTC)). We \*only\* use the XSLTC part of Xalan; We use  
the source from the packages that are part of the XSLTC sources.

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation.,  
<http://www.lotus.com>.
- software copyright (c) 2001-2002, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation., <http://www.ibm.com>.
- voluntary contributions made by Ovidiu Predescu ([ovidiu@cup.hp.com](mailto:ovidiu@cup.hp.com)) on behalf of the  
Apache Software Foundation and was originally developed at Hewlett Packard Company.

</pre>

### Apache 2.0 License

<pre>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control,  
are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare  
Derivative Works of,  
publicly display, publicly perform, sublicense, and distribute the  
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian  
Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software. The portions of JLex output which are hard-coded into the JLex source code are (naturally) covered by this same license.

</pre>

```
Little Color Management System (LCMS) v2.14
```

```
LCMS License
```

<pre>

README.1ST file information

LittleCMS core is released under MIT License

-----

Little CMS

Copyright (c) 1998-2022 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN

NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----

The below license applies to the following files:

liblcms/cmssm.c

Copyright 2001, softSurfer (www.softsurfer.com)

This code may be freely used and modified for any purpose providing that this copyright notice is included with it. SoftSurfer makes no warranty for this code, and cannot be held liable for any real or imagined damage resulting from its use. Users of this code must verify correctness for their application.

</pre>

### AUTHORS File Information

^^^

Main Author

-----

Marti Maria

Contributors

-----

Bob Friesenhahn

Kai-Uwe Behrmann

Stuart Nixon

Jordi Vilar

Richard Hughes

Auke Nauta

Chris Evans (Google)

Lorenzo Ridolfi

Robin Watts (Artifex)

Shawn Pedersen

Andrew Brygin

Samuli

Suominen

Florian Hch

Aurelien Jarno

Claudiu Cebuc  
Michael Vhrel (Artifex)  
Michal Cihar  
Daniel Kaneider  
Mateusz Jurczyk (Google)  
Paul Miller  
Sbastien Lon  
Christian Schmitz  
XhmikosR  
Stanislav Brabec (SuSe)  
Leonhard Gruenschloss (Google)  
Patrick Noffke  
Christopher James Halse Rogers  
John Hein  
Thomas Weber (Debian)  
Mark Allen  
Noel Carboni  
Sergei Trofimovic  
Philipp Knechtges

Special Thanks

-----

Artifex software  
AlienSkin software  
Jan Morovic  
Jos Vernon (WebSupergoo)  
Harald Schneider (Maxon)  
Christian Albrecht  
Dimitrios Anastassakis  
Lemke Software  
Tim Zaman

...

## GIFLIB v5.2.1

### GIFLIB License

...

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:



The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<https://sourceforge.net/p/giflib/code/ci/master/tree/openbsd-reallocarray.c>

Copyright (c) 2008 Otto Moerbeek <otto@drijf.net>

SPDX-License-Identifier: MIT

## CUP Parser Generator for Java v 0.11b

### CUP Parser Generator License

---

Copyright 1996-2015 by Scott Hudson, Frank Flannery, C. Scott Ananian, Michael Petter

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

---

---

---

This is an open source license. It is also GPL-Compatible (see entry for "Standard ML of New Jersey"). The portions of CUP output which are hard-coded into the CUP source code are (naturally) covered by this same license, as is the CUP runtime code linked with the generated parser.

---

## The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also,

for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced

by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its

terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically

performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source

code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by

third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later

version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

How  
to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a

pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into



proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#### "CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are subject to the following clarification and special exception to the GPL, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module

which is not derived from or based on this library. If

you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

## Apache Commons Byte Code Engineering Library (BCEL) Version 6.5.0

### Apache Commons BCEL Notice

<pre>

Apache Commons BCEL  
Copyright 2004-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

</pre>

### Apache 2.0 License

<pre>

Apache License  
Version 2.0, January 2004

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity.

For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

</pre>

## Unicode Common Local Data Repository (CLDR) v39

### CLDR License

^^^

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2021 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal

in the Data Files or Software  
without restriction, including without limitation the rights to use,  
copy, modify, merge, publish, distribute, and/or sell copies of  
the Data Files or Software, and to permit persons to whom the Data Files  
or Software are furnished to do so, provided that either  
(a) this copyright and permission notice appear with all copies  
of the Data Files or Software, or  
(b) this copyright and permission notice appear in associated  
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF  
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS  
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL  
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,  
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER  
TORTIOUS ACTION,

ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder  
shall not be used in advertising or otherwise to promote the sale,  
use or other dealings in these Data Files or Software without prior  
written authorization of the copyright holder.

----- Terms of Use -----  
-

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the [Unicode Privacy Policy](#).

Unicode Copyright

Copyright 1991-2021 Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

Unicode

Software ("SOFTWARE") includes any source code published in the Unicode Standard  
or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

#### Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions

(Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

#### Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government

is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

#### Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

**EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR**

**SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.**

#### Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

#### Trademarks & Logos



The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

Miscellaneous

**Jurisdiction and Venue.** This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and

damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

**Modification by Unicode, Inc.** Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

**Taxes.** The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

**Severability.** If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

Entire

Agreement. This Agreement constitutes the entire agreement between the parties.

...

## ASM Bytecode Manipulation Framework v8.0.1

### ASM License

<pre>

Copyright (c) 2000-2011 France Tlcom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## Dynalink v.5

### Dynalink License

<pre>

Copyright (c) 2009-2013, Attila Szegedi

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## Independent JPEG Group: JPEG release 6b

### JPEG License

...

\*\*\*\*\*

Copyright (C) 1991-1998, Thomas G. Lane.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group.

IJG is not affiliated with the official ISO JPEG standards committee.

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.  
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that "The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

\*\*\*\*\*  
^^^

## International Components for Unicode (ICU4J) v67.1

### ICU4J License  
^^^

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2020 Unicode, Inc. All rights reserved.  
Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either  
(a) this copyright and permission notice appear with all copies of the Data Files or Software, or  
(b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

-----

#### Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

#### 1. ICU License - ICU 1.8.1 to ICU 57.1

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of

the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

From: <https://www.unicode.org/copyright.html>:

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the [Unicode Privacy Policy](#).

Unicode Copyright

Copyright 1991-2020

Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

Unicode Software ("SOFTWARE") includes any source code published in the Unicode Standard or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

#### Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions (Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

#### Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

#### Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

#### Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

## Trademarks & Logos

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

## Miscellaneous

**Jurisdiction and Venue.** This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

**Modification by Unicode, Inc.** Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

**Taxes.** The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

**Severability.** If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

**Entire Agreement.** This Agreement constitutes the entire agreement between the parties.

...

The copyrights in this software and any visual or audio work distributed with the software belong to Azul Systems, Inc. and those included in all other notice files either listed in the readme file or contained in any other included notice files with this distribution. All rights are reserved. Installation of this software and any Azul software bundled with or derived from this software is licensed only in accordance with these terms.

Provided you have not received the software directly from Azul and have already agreed to the terms of a separate license agreement, by installing, using or distributing this software you, on your own behalf and on behalf of your employer or principal, agree to be bound by these terms. If you do not agree to any of these terms, you may not use, copy, transmit, distribute nor install this software.

The software is developed and owned by Azul and/or any of its affiliates, subsidiaries or respective suppliers and licensors. The software also includes certain

software components and materials ("Open Source Materials") of third parties ("Third Party Licensors") licensed under certain licenses ("Open Source Licenses").



Nothing in this Agreement limits your rights under, or grants you rights that supersede the terms and conditions of any applicable Open Source Licenses. You must review the Open Source Licenses located at [http://www.azulsystems.com/license/zulu\\_third\\_party\\_licenses.html](http://www.azulsystems.com/license/zulu_third_party_licenses.html) to understand your rights under them. In the event that no such third party license agreements exist, the restrictions contained in this Agreement shall apply.

With respect to the Open Source Licenses the following shall apply:

- 1) You hereby acknowledge and agree that you will be licensing any Open Source Materials directly from the applicable Third Party Licensors including the right to use such Third Party Materials in connection with the software
- 2) In the event of any inconsistencies or conflicting provisions of the Open Source Licenses and the provisions of this Agreement, the provisions of the Open Source Licenses shall prevail.

THIS SOFTWARE IS PROVIDED BY AZUL "AS-IS" AND WITHOUT EXPRESS OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL AZUL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICE, LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY WHETHER IN CONTRACT, STRICT LIABILITY OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Mesa 3-D Graphics Library v21.0.3

### Mesa License

...

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Attention, Contributors

When contributing to the Mesa project you must agree to the licensing terms of the component to which you're contributing.

The following section lists the primary components of the Mesa distribution and their respective licenses.

Mesa Component Licenses

Component	Location	License
Main Mesa code	src/mesa/	MIT
Device drivers	src/mesa/drivers/*	MIT, generally
Gallium code	src/gallium/	MIT
Ext headers	GL/glext.h	Khronos
	GL/gltext.h	Khronos
	GL/wgltext.h	Khronos
	KHR/khrplatform.h	Khronos

\*\*\*\*\*

----  
include/GL/gl.h :

Mesa 3-D  
graphics library

Copyright (C) 1999-2006 Brian Paul All Rights Reserved.  
Copyright (C) 2009 VMware, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*

----

```
include/GL/glex.h
include/GL/glxext.h
include/GL/wglxext.h :
```

Copyright (c) 2013 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

\*\*\*\*\*

----

```
include/KHR/khrplatform.h :
```

Copyright (c) 2008 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

\*\*\*\*\*

...

## Mozilla Public Suffix List

### Public Suffix Notice

...

You are receiving a copy of the Mozilla Public Suffix List in the following file: <java-home>/lib/security/public\_suffix\_list.dat. The terms of the Oracle license do NOT apply to this file; it is licensed under the Mozilla Public License 2.0, separately from the Oracle programs you receive. If you do not wish to use the Public Suffix List, you may remove the <java-home>/lib/security/public\_suffix\_list.dat file.

The Source Code of this file is available under the Mozilla Public License, v. 2.0 and is located at [https://raw.githubusercontent.com/publicsuffix/list/3c213aab32b3c014f171b1673d4ce9b5cd72bf1c/public\\_suffix\\_list.dat](https://raw.githubusercontent.com/publicsuffix/list/3c213aab32b3c014f171b1673d4ce9b5cd72bf1c/public_suffix_list.dat).

If a copy of the MPL was not distributed with this file, you can obtain one at <https://mozilla.org/MPL/2.0/>.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

...

### MPL v2.0

...

## Mozilla Public License Version 2.0

=====

### 1. Definitions

-----

#### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

#### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

#### 1.3. "Contribution"

means Covered Software of a particular Contributor.

#### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

#### 1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

#### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

#### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. "License"

means this document.

#### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and

all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent

claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect,

to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

-----

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- (b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

-----

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms

of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3.



## Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

-----

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must:

(a) comply with

the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

-----

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

\*\*\*\*\*

\*

\*

\* 6. Disclaimer of Warranty

\*

\* -----

\*

\*

\*

\* Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

essential part of this License. No use of any Covered Software is \*  
\* authorized under this License except under this disclaimer. \*  
\* \*  
\*\*\*\*\*

\*\*\*\*\*

\* \*  
\* 7. Limitation of Liability \*  
\* ----- \*

\* \*  
\* Under no circumstances and under no legal theory, whether tort \*  
\* (including negligence), contract, or otherwise, shall any \*  
\* Contributor, or anyone who distributes Covered Software as \*  
\* permitted above, be liable to You for any direct, indirect, \*  
\* special, incidental, or consequential damages of any  
character \*  
\* including, without limitation, damages for lost profits, loss of \*  
\* goodwill, work stoppage, computer failure or malfunction, or any \*  
\* and all other commercial damages or losses, even if such party \*  
\* shall have been informed of the possibility of such damages. This \*  
\* limitation of liability shall not apply to liability for death or \*  
\* personal injury resulting from such party's negligence to the \*  
\* extent applicable law prohibits such limitation. Some \*  
\* jurisdictions do not allow the exclusion or limitation of \*  
\* incidental or consequential damages, so this exclusion and \*  
\* limitation may not apply to You. \*  
\* \*  
\*\*\*\*\*

### 8. Litigation -----

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

### 9. Miscellaneous -----

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter

shall not be used to construe this License against a Contributor.

## 10. Versions of the License

-----

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License.

Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

-----

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

-----  
This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

...

## 1.161 jsr305 3.0.2

### 1.161.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) 2005 Brian Goetz
 * Released under the Creative Commons Attribution License
 * (http://creativecommons.org/licenses/by/2.5)
 * Official home: http://www.jcip.net
 */
```

Found in path(s):

```
* /opt/cola/permits/1656638364_1682593601.2844107/0/jsr305-3-0-2-sources-9-jar/javax/annotation/concurrent/ThreadSafe.java
* /opt/cola/permits/1656638364_1682593601.2844107/0/jsr305-3-0-2-sources-9-jar/javax/annotation/concurrent/NotThreadSafe.java
* /opt/cola/permits/1656638364_1682593601.2844107/0/jsr305-3-0-2-sources-9-jar/javax/annotation/concurrent/Immutable.java
* /opt/cola/permits/1656638364_1682593601.2844107/0/jsr305-3-0-2-sources-9-jar/javax/annotation/concurrent/GuardedBy.java
```

## 1.162 xml-apis 1.4.01

### 1.162.1 Available under license :

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache xml-commons xml-apis ==
== distribution. ==
=====
```

Apache XML Commons XML APIs  
Copyright 1999-2009 The Apache Software Foundation.

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2000 World Wide Web Consortium, <http://www.w3.org>

xml-commons/java/external/LICENSE.sax.txt \$Id: LICENSE.sax.txt 225954 2002-01-31 23:26:48Z curcuru \$

This license came from: <http://www.megginson.com/SAX/copying.html>

However please note future versions of SAX may be covered  
under <http://saxproject.org/?selected=pd>

This page is now out of date -- see the new SAX site at  
<http://www.saxproject.org/> for more up-to-date  
releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, [david@megginson.com](mailto:david@megginson.com)

2000-05-05

xml-commons/java/external/LICENSE.dom-software.txt \$Id: LICENSE.dom-software.txt 734314 2009-01-14 03:33:27Z mrglavas \$

This license came from: <http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.zip>  
(COPYRIGHT.html)

#### W3C SOFTWARE NOTICE AND LICENSE

Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University).

All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the

W3C Software Copyright Notice and License could  
be found at <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.

Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent



to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

xml-commons/java/external/LICENSE.dom-documentation.txt \$Id: LICENSE.dom-documentation.txt 226215 2005-06-03 22:49:13Z mrglavas \$

This license came from: <http://www.w3.org/Consortium/Legal/copyright-documents-20021231>

### W3C DOCUMENT LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [ \$date-of-document ] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.  
<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in

the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

-----  
This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to [site-policy@w3.org](mailto:site-policy@w3.org).

Joseph Reagle <[site-policy@w3.org](mailto:site-policy@w3.org)>

Last revised by Reagle \$Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) \$

## 1.163 servlet-api 3.0.1

### 1.163.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

#### 1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8.

Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.

Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

### 2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

### 2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

### 3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

### 3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

### 3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

### 3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### 3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

## 4. Versions of the License.



#### 4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

#### 4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the

License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

#### 4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

### 5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES

THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

### 6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software

against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

## 7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS

DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

## 8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

## 9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

## 10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

#### NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software,

we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer

you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code

for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

## Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## "CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that



accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but

you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

## 1.164 perfmark-api 0.17.0

### 1.164.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 Carl Mastrangelo
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1656707133_1682600610.4687467/0/perfmark-api-0-17-0-sources-1-jar/io/perfmark/Impl.java
* /opt/cola/permits/1656707133_1682600610.4687467/0/perfmark-api-0-17-0-sources-1-
jar/io/perfmark/PerfMark.java
* /opt/cola/permits/1656707133_1682600610.4687467/0/perfmark-api-0-17-0-sources-1-jar/io/perfmark/package-
info.java
*
/opt/cola/permits/1656707133_1682600610.4687467/0/perfmark-api-0-17-0-sources-1-jar/io/perfmark/Link.java
* /opt/cola/permits/1656707133_1682600610.4687467/0/perfmark-api-0-17-0-sources-1-jar/io/perfmark/Tag.java
```

# 1.165 jackson-annotations 2.15.0

## 1.165.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0  
To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

## 1.166 jackson-dataformat-yaml 2.15.0

## 1.166.1 Available under license :

This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License").  
See the License for details about distribution rights, and the specific rights regarding derivative works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.  
It is currently developed by a community of developers.

### ## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

### ## Licensing

Jackson components are licensed under Apache (Software) License, version 2.0, as per accompanying LICENSE file.

### ## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

## 1.167 jackson-databind 2.15.0

### 1.167.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but



excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0 To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

# 1.168 disruptor 2.11.2.3

## 1.168.1 Available under license :

Apache Log4j Core  
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java  
Copyright 2005-2006 Tim Fennell

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.



You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.169 openjdk 17.0.7

## 1.169.1 Available under license :

```
c-libutl 20160225
```

```
c-libutl License

```

This software is distributed under the terms of the BSD license.

```
== BSD LICENSE =====
```

(C) 2009 by Remo Dentato (rdentato@gmail.com)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<http://opensource.org/licenses/bsd-license.php>

...

## Apache Xerces v2.12.2

### Apache Xerces Notice

<pre>

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
=====
```

Apache Xerces Java  
Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

</pre>

### Apache 2.0 License

<pre>

```
 Apache License
 Version 2.0, January 2004
 http://www.apache.org/licenses/
```

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

</pre>

## JLine v3.20.0

### JLine License

<pre>

Copyright (c) 2002-2018, the original author or authors.

All rights reserved.

<https://opensource.org/licenses/BSD-3-Clause>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

Copyright (c) 1999-2003 David Corcoran <corcoran@musclecard.com>

Copyright (c) 2001-2011 Ludovic Rousseau <ludovic.rousseau@free.fr>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN

NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some files are under GNU GPL v3 or any later version

- doc/example/pcsc\_demo.c
- the files in src/spy/
- the files in UnitaryTests/

Copyright (C) 2003-2014 Ludovic Rousseau

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Files src/auth.c and src/auth.h are:



\* Copyright (C) 2013 Red Hat  
\*  
\* All rights reserved.  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\*  
\* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.  
\*  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\*  
\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
\* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
\* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS  
\* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
\* COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,  
\* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS  
\* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED  
\* AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
\* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF  
\* THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH  
\* DAMAGE.  
\*  
\* Author: Nikos Mavrogiannopoulos <nmav@redhat.com>

Files src/simclist.c and src/simclist.h are:

\* Copyright (c) 2007,2008,2009,2010,2011 Mij <mij@bitchx.it>  
\*  
\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby  
\* granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
\* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
\* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
\* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## xwd v1.0.7

### xwd utility

<pre>

This is the copyright for the files in src/java.desktop/unix/native/libawt\_xawt:  
list.h, multiVis.h, wsutils.h, list.c, multiVis.c

Copyright 1994 Hewlett-Packard Co.  
Copyright 1996, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

</pre>

## IAIK (Institute for Applied Information Processing and Communication) PKCS#11 wrapper files v1

### IAIK License

<pre>

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of Technology."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## The FreeType Project: Freetype v2.12.1

### FreeType Notice

...

FreeType comes with two licenses from which you can choose the one which fits your needs best.

The FreeType License (FTL) is the most commonly used one. It is a BSD-style license with a credit clause and thus compatible with the GNU Public License (GPL) version 3, but not with the GPL version 2.

The GNU General Public License (GPL), version 2. Use it for all projects which use the GPLv2 also, or which need a license

compatible to the GPLv2.

...

### FreeType License

...

Copyright (C) 1996-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
 Copyright (C) 2007-2022 by Dereg Clegg and Michael Toftdal.  
 Copyright (C) 1996-2022 by Just van Rossum, David Turner, Robert Wilhelm, and Werner Lemberg.  
 Copyright (C) 2004-2022 by Masatake YAMATO and Redhat K.K.  
 Copyright (C) 2007-2022 by Derek Clegg and Michael Toftdal.  
 Copyright (C) 2007-2022 by David Turner.  
 Copyright (C) 2022 by David Turner,  
 Robert Wilhelm, Werner Lemberg, and Moazin Khatti.  
 Copyright (C) 2007-2022 by Rahul Bhalerao <rahul.bhalerao@redhat.com>, <b.rahul.pm@gmail.com>.  
 Copyright (C) 2008-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and suzuki toshiya.  
 Copyright (C) 2019-2022 by Nikhil Ramakrishnan, David Turner, Robert Wilhelm, and Werner Lemberg.  
 Copyright (C) 2009-2022 by Oran Agra and Mickey Gabel.  
 Copyright (C) 2004-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and George Williams.  
 Copyright (C) 2004-2022 by Masatake YAMATO, Red Hat K.K.,  
 Copyright (C) 2003-2022 by Masatake YAMATO, Redhat K.K.,  
 Copyright (C) 2013-2022 by Google, Inc.  
 Copyright (C) 2018-2022 by David Turner, Robert Wilhelm, Dominik Rttches, and Werner Lemberg.  
 Copyright (C) 2005-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
 Copyright 2013 by Google, Inc.

The FreeType Project LICENSE

-----

2006-Jan-27

Copyright 1996-2002, 2006 by  
 David Turner, Robert Wilhelm, and Werner Lemberg

Introduction

=====

The FreeType Project is distributed in several archive packages;  
 some of them may contain, in addition to the FreeType font engine,  
 various tools and contributions which rely on, or relate to, the  
 FreeType Project.

This license applies to all files found in such packages, and

which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. (^ as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. (^ royalty-free' usage)
- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (^ credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
""
Portions of this software are copyright <year> The FreeType
Project (www.freetype.org). All rights reserved.
""
```

Please replace <year> with the value from the FreeType version you actually use.

## Legal Terms

=====

### 0. Definitions

-----

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally

distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

'You' refers to the licensee, or person using the project, where 'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive.

If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

#### 1. No Warranty

-----

THE FREETYPE PROJECT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

#### 2. Redistribution

-----

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered,

original files must be preserved in all copies of source files.

- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

-----

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

### 4. Contacts

-----

There are two mailing lists related to FreeType:

- o [freetype@nongnu.org](mailto:freetype@nongnu.org)

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution.

If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- o [freetype-devel@nongnu.org](mailto:freetype-devel@nongnu.org)

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

<http://www.freetype.org>

...

### GPL v2

...

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it.

By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether



gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally,  
any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any

such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to

the Free Software

Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
...
```

```
Additional Freetype Attributions
...
```

```

```

The below license applies to the following files:

```
libfreetype/src/psaux/psarrst.c
libfreetype/src/psaux/psarrst.h
libfreetype/src/psaux/psblues.c
libfreetype/src/psaux/psblues.h
libfreetype/src/psaux/pserror.c
```

libfreetype/src/psaux/pserror.h  
libfreetype/src/psaux/psfixed.h  
libfreetype/src/psaux/psfont.c  
libfreetype/src/psaux/psfont.h  
libfreetype/src/psaux/psft.c  
libfreetype/src/psaux/psft.h  
libfreetype/src/psaux/psglue.h  
libfreetype/src/psaux/pshints.c  
libfreetype/src/psaux/pshints.h  
libfreetype/src/psaux/psintrap.c  
libfreetype/src/psaux/psintrap.h  
libfreetype/src/psaux/psread.c  
libfreetype/src/psaux/psread.h  
libfreetype/src/psaux/psstack.c  
libfreetype/src/psaux/psstack.h  
libfreetype/src/psaux/pstypes.h

#### Copyright

2006-2014 Adobe Systems Incorporated.

This software, and all works of authorship, whether in source or object code form as indicated by the copyright notice(s) included herein (collectively, the "Work") is made available, and may only be used, modified, and distributed under the FreeType Project License, LICENSE.TXT. Additionally, subject to the terms and conditions of the FreeType Project License, each contributor to the Work hereby grants to any individual or legal entity exercising permissions granted by the FreeType Project License and this section (hereafter, "You" or "Your") a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such contributor that are necessarily infringed by their contribution(s) alone or by combination of their contribution(s) with the Work to which such contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

By using, modifying, or distributing the Work you indicate that you have read and understood the terms and conditions of the FreeType Project License as well as those provided in this section, and you accept



them fully.

...

### MIT License

...

-----  
The below license applies to the following files:

libfreetype/include/freetype/internal/fthash.h

libfreetype/src/base/fthash.c

Copyright 2000 Computing Research Labs, New Mexico State University

Copyright 2001-2015

Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

...

## Harfbuzz v4.4.1

### Harfbuzz License

<https://github.com/harfbuzz/harfbuzz/blob/4.4.1/COPYING>

<pre>

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow.

For parts of HarfBuzz that are licensed under different licenses see individual files names COPYING in subdirectories where applicable.

Copyright 2010-2022 Google, Inc.  
Copyright 2018-2020 Ebrahim Byagowi  
Copyright 2019-2020 Facebook, Inc.  
Copyright 2012-2015 Mozilla Foundation.  
Copyright 2011 Codethink Limited  
Copyright 2008-2010 Nokia Corporation and/or its subsidiary(-ies)  
Copyright 2009 Keith Stribley  
Copyright 2009 Martin Hosken and SIL International  
Copyright 2007 Chris Wilson  
Copyright 2005-2022 Behdad Esfahbod  
Copyright 2005 David Turner  
Copyright 2004-2013 Red Hat, Inc.  
Copyright 1998-2004 David Turner and Werner Lemberg  
Copyright 2016 Elie Roux <elie.roux@telecom-bretagne.eu>  
Copyright 2018-2019 Adobe Inc.  
Copyright 2018 Khaled Hosny  
Copyright 2016 Igalia S.L.

For  
full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

All source  
code, except for one section, is licensed as above. The one  
exception is licensed with a slightly different MIT variant:  
The contents of this directory are licensed under the following terms:

-----  
The below license applies to the following files:

libharfbuzz/hb-ucd.cc

Copyright (C) 2012 Grigori Goronzy <greg@kinoho.net>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

</pre>

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## PC/SC Lite v1.9.5

### PC/SC Lite License

<pre>

The main parts of the code are using the BSD-like licence bellow:

Copyright (c) 1999-2003 David Corcoran <corcoran@linuxnet.com>

Copyright (c) 2001-2011 Ludovic Rousseau <ludovic.rousseau@free.fr>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Changes to this license can be made only by the copyright author with explicit written consent.

THIS

SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some source code files are using other licences. See the [COPYING](<https://salsa.debian.org/rousseau/PCSC/-/blob/master/COPYING>) file for details.

</pre>

```
The Unicode Standard, Unicode Character Database, Version 13.0.0
```

```
Unicode Character Database
```

```
...
```

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s  
Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement.  
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S  
DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),  
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE  
TERMS AND CONDITIONS OF THIS AGREEMENT.  
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE  
THE DATA FILES OR SOFTWARE.

## COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2020 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

...

## jQuery UI v1.12.1

### jQuery UI License

...

Copyright jQuery Foundation and other contributors, <https://jquery.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/jquery-ui>

The following license applies to all parts of this software except as

documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code contained within the demos directory.

CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

====

All files located in the node\_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

...

This is the copyright file

```
zlib v1.2.13
```

```
zlib License
```

```
<pre>
```

Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied

warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly      Mark Adler  
jloup@gzip.org      madler@alumni.caltech.edu

```
</pre>
jpackage test license file (just some sample text).
Eastman Kodak Company: Portions of color management and imaging software
```

```
Eastman Kodak Notice
<pre>
Portions Copyright Eastman Kodak Company 1991-2003
</pre>
Thai Dictionary
```

```
Thai Dictionary License
<pre>
```

Copyright (C) 1982 The Royal Institute, Thai Royal Government.

Copyright (C) 1998 National Electronics and Computer Technology Center,  
National Science and Technology Development Agency,  
Ministry of Science Technology and Environment,  
Thai Royal Government.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

## Apache Santuario v2.3.0

### Apache Santuario Notice

<pre>

Apache Santuario - XML Security for Java  
Copyright 1999-2021 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

It was originally based on software copyright (c) 2001, Institute for  
Data Communications Systems, <<http://www.nue.et-inf.uni-siegen.de/>>.

The development of this software was partly funded by the European  
Commission in the <WebSig> project in the ISIS Programme.

</pre>

### Apache 2.0 License

<pre>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all



other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except  
in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

</pre>

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files: \*

Copyright: APPLICATION\_COPYRIGHT

License: APPLICATION\_LICENSE\_TEXT

#### ADDITIONAL INFORMATION ABOUT LICENSING

Certain files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception.

Note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0 and may include FreeType. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result.

Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Failing to distribute notices associated with some files may also create unexpected legal consequences.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications

to this package which may subsequently be redistributed  
and/or involve the use of third party software.  
Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.  
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.

This code is free software; you can redistribute it and/or modify it  
under the terms of the GNU General Public License version 2 only, as  
published by the Free Software Foundation. Oracle designates this  
particular file as subject to the "Classpath" exception as provided  
by Oracle in the LICENSE file that accompanied this code.

This code is distributed in the hope that it will be useful, but WITHOUT  
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License  
version 2 for more details (a copy is included in the LICENSE file that  
accompanied this code).

You should have received a copy of the GNU General Public License version  
2 along with this work; if not, write to the Free Software Foundation,  
Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

Please contact

Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA  
or visit [www.oracle.com](http://www.oracle.com) if you need additional information or have any  
questions.

## OASIS PKCS #11 Cryptographic Token Interface v3.0

### OASIS PKCS #11 Cryptographic Token Interface License

<pre>

Copyright OASIS Open 2020. All Rights Reserved.

All capitalized terms in the following text have the meanings  
assigned to them in the OASIS Intellectual Property Rights Policy (the  
"OASIS IPR Policy"). The full Policy may be found at the OASIS website:  
[<http://www.oasis-open.org/policies-guidelines/ipr>]

This document and translations of it may be copied and furnished to  
others, and derivative works that comment on or otherwise explain it or  
assist in its implementation may be prepared, copied, published, and  
distributed, in whole or in part, without restriction of any kind,  
provided that the above copyright notice and this section are included  
on all such copies and derivative works. However, this document itself  
may not be modified in any way, including by removing the copyright  
notice or references to OASIS, except as needed for the purpose of  
developing any

document or deliverable produced by an OASIS Technical  
Committee (in which case the rules applicable to copyrights, as set

forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of

intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

</pre>

## DOM Level 3 Core Specification v1.0

### W3C Software Notice

<pre>

Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

</pre>

### W3C License

<pre>

W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.



3. Notice of any changes or modifications to the files, including the date changes were made.  
(We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

---

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to [site-policy@w3.org](mailto:site-policy@w3.org).

```
</pre>
Cryptix AES v3.2.0

Cryptix General License
<pre>
```

Cryptix General License

Copyright (c) 1995-2005 The Cryptix Foundation Limited.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are

met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
</pre>
libpng v1.6.38
```

```
libpng License
<pre>
```

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE  
=====

PNG Reference Library License version 2  
-----

Copyright (c) 1995-2022 The PNG Reference Library Authors.  
Copyright (c) 2018-2022 Cosmin Truta  
Copyright (c) 1998-2018 Glenn Randers-Pehrson  
Copyright (c) 1996-1997 Andreas Dilger  
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising

from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

-----  
libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Mans Rullgard  
Cosmin Truta  
Gilles Vollant  
James Yu  
Mandar Sahastrabuddhe  
Google Inc.  
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is

with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000

Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of

merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

#### TRADEMARK

=====

The name "libpng" has not been registered by the Copyright owners as a trademark in any jurisdiction. However, because libpng has been distributed and maintained world-wide, continually since 1995, the Copyright owners claim "common-law trademark protection" in any jurisdiction where common-law trademark is recognized.

</pre>

### AUTHORS File Information

...

#### PNG REFERENCE LIBRARY AUTHORS

=====

This is the list of PNG Reference Library ("libpng") Contributing Authors, for copyright and licensing purposes.

- \* Andreas Dilger
- \* Cosmin Truta
- \* Dave Martindale
- \* Eric S. Raymond

- \* Gilles Vollant
- \* Glenn Randers-Pehrson
- \* Greg Roelofs
- \* Guy Eric Schalnat
- \* James Yu
- \* John Bowler
- \* Kevin Bracey
- \* Magnus Holmgren
- \* Mandar Sahastrabudde
- \* Mans Rullgard
- \* Matt Sarett
- \* Mike Klein
- \* Pascal Massimino
- \* Paul Schmidt
- \* Qiang Zhou
- \* Sam Bushell
- \* Samuel Williams
- \* Simon-Pierre Cadieux
- \* Tim Wegner
- \* Tom Lane
- \* Tom Tanner
- \* Vadim Barkov
- \* Willem van Schaik
- \* Zhijie Liang
- \* Arm Holdings
  - Richard Townsend
- \* Google Inc.
  - Matt Sarett
  - Mike Klein
  - Dan Field
  - Sami Boukortt

The build projects, the build scripts, the test scripts, and other files in the "ci", "projects", "scripts" and "tests" directories, have other copyright owners, but are released under the libpng license.

Some files in the "contrib" directory, and some tools-generated files that are distributed with libpng, have other copyright owners, and are released under other open source licenses.

...

## jopt-simple v5.0.4

### MIT License

<pre>

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

## jQuery v3.6.1

### jQuery License

...

jQuery v 3.6.1

Copyright OpenJS Foundation and other contributors, <https://openjsf.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*

The jQuery JavaScript Library v3.6.1 also includes Sizzle.js

Sizzle.js includes the following license:

Copyright JS Foundation and other contributors, <https://js.foundation/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/sizzle>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node\_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

\*\*\*\*\*

...

## Apache Xalan v2.7.2



### Apache Xalan Notice

<pre>

```
=====
====
== NOTICE file corresponding to the section 4d of the Apache License, Version 2.0, ==
== in this case for the Apache Xalan distribution. ==
=====
====
```

This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).

Specifically, we only include the XSLTC portion of the source from the Xalan distribution.  
The Xalan project has two processors: an interpretive one (Xalan Interpretive) and a  
compiled one (The XSLT Compiler (XSLTC)). We \*only\* use the XSLTC part of Xalan; We use  
the source from the packages that are part of the XSLTC sources.

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation.,  
<http://www.lotus.com>.
- software copyright (c) 2001-2002, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation., <http://www.ibm.com>.
- voluntary contributions made by Ovidiu Predescu ([ovidiu@cup.hp.com](mailto:ovidiu@cup.hp.com)) on behalf of the  
Apache Software Foundation and was originally developed at Hewlett Packard Company.

</pre>

### Apache 2.0 License

<pre>

```
 Apache License
 Version 2.0, January 2004
 http://www.apache.org/licenses/
```

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control,

are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare  
Derivative Works of,  
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative

Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to

in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian  
Permission to use, copy, modify, and distribute this software and  
its documentation for any purpose and without fee is hereby granted,  
provided that the above copyright notice appear in all copies and that  
both the copyright notice and this permission notice and warranty  
disclaimer appear in supporting documentation, and that the name of  
the authors or their employers not be used in advertising or publicity  
pertaining to distribution of the software without specific, written  
prior permission.

The authors and their employers disclaim all warranties with regard to  
this software, including all implied warranties of merchantability  
and

fitness. In no event shall the authors or their employers be liable for  
any special, indirect or consequential damages or any damages whatsoever  
resulting from loss of use, data or profits, whether in an action of  
contract, negligence or other tortious action, arising out of or in  
connection with the use or performance of this software. The portions of  
JLex output which are hard-coded into the JLex source code are (naturally)  
covered by this same license.

</pre>

```
Little Color Management System (LCMS) v2.14
```

```
LCMS License
```

<pre>

README.1ST file information

LittleCMS core is released under MIT License

-----

Little CMS

Copyright (c) 1998-2022 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated documentation files (the  
"Software"), to deal in the Software without restriction, including  
without limitation the rights to use, copy, modify, merge, publish,  
distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN

NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
The below license applies to the following files:  
liblcms/cmssm.c

Copyright 2001, softSurfer (www.softsurfer.com)

This code may be freely used and modified for any purpose providing that this copyright notice is included with it. SoftSurfer makes no warranty for this code, and cannot be held liable for any real or imagined damage resulting from its use. Users of this code must verify correctness for their application.

</pre>

### AUTHORS File Information  
^^^

Main Author

-----  
Marti Maria

Contributors

-----  
Bob Friesenhahn  
Kai-Uwe Behrmann  
Stuart Nixon  
Jordi Vilar  
Richard Hughes  
Auke Nauta  
Chris Evans (Google)

Lorenzo Ridolfi  
Robin Watts (Artifex)  
Shawn Pedersen  
Andrew Brygin  
Samuli  
Suominen  
Florian Hch  
Aurelien Jarno  
Claudiu Cebuc  
Michael Vhrel (Artifex)  
Michal Cihar  
Daniel Kaneider  
Mateusz Jurczyk (Google)  
Paul Miller  
Sbastien Lon  
Christian Schmitz  
XhmikosR  
Stanislav Brabec (SuSe)  
Leonhard Gruenschloss (Google)  
Patrick Noffke  
Christopher James Halse Rogers  
John Hein  
Thomas Weber (Debian)  
Mark Allen  
Noel Carboni  
Sergei Trofimovic  
Philipp Knechtges

Special Thanks

-----

Artifex software  
AlienSkin software  
Jan Morovic  
Jos Vernon (WebSupergoo)  
Harald Schneider (Maxon)  
Christian Albrecht  
Dimitrios Anastassakis  
Lemke Software  
Tim Zaman

...

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

=====

PNG Reference Library License version 2

-----



- \* Copyright (c) 1995-2022 The PNG Reference Library Authors.
- \* Copyright (c) 2018-2022 Cosmin Truta.
- \* Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
- \* Copyright (c) 1996-1997 Andreas Dilger.
- \* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted  
to use, copy, modify, and distribute  
this software, or portions hereof, for any purpose, without fee,  
subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

-----  
libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are  
Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are  
derived from libpng-1.0.6, and are distributed according to the same  
disclaimer

and license as libpng-1.0.6 with the following individuals  
added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Mans Rullgard  
Cosmin Truta  
Gilles Vollant

James Yu  
Mandar Sahastrabuddhe  
Google Inc.  
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

All images in this directory are copyright 1995 by Jeff Dinkins.  
## GIFLIB v5.2.1

### GIFLIB License  
...

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<https://sourceforge.net/p/giflib/code/ci/master/tree/openbsd-reallocarray.c>

Copyright (c) 2008 Otto Moerbeek <otto@drijf.net>

SPDX-License-Identifier: MIT

## CUP Parser Generator for Java v 0.11b

### CUP Parser Generator License

...

Copyright 1996-2015 by Scott Hudson, Frank Flannery, C. Scott Ananian, Michael Petter

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

...

---

...

This is an open source license. It is also GPL-Compatible (see entry for

"Standard ML of New Jersey"). The portions of CUP output which are hard-coded into the CUP source code are (naturally) covered by this same license, as is the CUP runtime code linked with the generated parser.  
^^^

All images in this directory are copyright 1995 by Jeff Dinkins.  
Unauthorized reproduction is prohibited.

For more information about Jeff's photographs, please see:

<http://www.theFixx.org/Jeff>

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also,

for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its

terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under

Section 2) in object code or executable form under the terms of Sections 1 and

2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source

code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or



any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In

such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free
software, and you are welcome to redistribute it under certain conditions;
type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here

is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General Public  
License instead of this License.

#### "CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are  
subject to the following  
clarification and special exception to the GPL, but  
only where Oracle has expressly included in the particular source file's header  
the words "Oracle designates this particular file as subject to the "Classpath"  
exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making  
a combined work based on this library. Thus, the terms and conditions of  
the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you  
permission to link this library with independent modules to produce an  
executable, regardless of the license terms of these independent modules,  
and to copy and distribute the resulting executable under terms of your  
choice, provided that you also meet, for each linked independent module,  
the terms and conditions of the license of that module. An independent  
module is a module

which is not derived from or based on this library. If  
you modify this library, you may extend this exception to your version of  
the library, but you are not obligated to do so. If you do not wish to do  
so, delete this exception statement from your version.

## Apache Commons Byte Code Engineering Library (BCEL) Version 6.5.0

### Apache Commons BCEL Notice

<pre>

Apache Commons BCEL  
Copyright 2004-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

</pre>

### Apache 2.0 License

<pre>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity.

For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution

notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty



or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

</pre>

## Unicode Common Local Data Repository (CLDR) v39

### CLDR License

...

## UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement.  
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),  
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT.  
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE

THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2021 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,

ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

----- Terms of Use -----

-

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the Unicode Privacy Policy.

Unicode Copyright

Copyright 1991-2021 Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

#### Unicode

Software ("SOFTWARE") includes any source code published in the Unicode Standard or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

#### Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions

(Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

#### Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government

is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

#### Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR

SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

#### Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

#### Trademarks & Logos

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

#### Miscellaneous

**Jurisdiction and Venue.** This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and

damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

**Modification by Unicode, Inc.** Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

**Taxes.** The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

**Severability.** If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

#### Entire

Agreement. This Agreement constitutes the entire agreement between the parties.

...

## ASM Bytecode Manipulation Framework v8.0.1

### ASM License

<pre>

Copyright (c) 2000-2011 France Tlcom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Oracle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Dynalink v.5

### Dynalink License

<pre>

Copyright (c) 2009-2013, Attila Szegedi

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## Independent JPEG Group: JPEG release 6b

### JPEG License

...

\*\*\*\*\*

Copyright (C) 1991-1998, Thomas G. Lane.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group.

IJG is not affiliated with the official ISO JPEG standards committee.

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.  
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis

of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that "The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

\*\*\*\*\*

...

## International Components for Unicode (ICU4J) v67.1

### ICU4J License

...

#### COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2020 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.



THE DATA FILES AND SOFTWARE ARE PROVIDED  
"AS IS", WITHOUT WARRANTY OF  
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT OF THIRD PARTY RIGHTS.  
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS  
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL  
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,  
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER  
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder  
shall not be used in advertising or otherwise to promote the sale,  
use or other dealings in these Data Files or Software without prior  
written authorization of the copyright holder.

-----

#### Third-Party Software Licenses

This section contains third-party software notices and/or additional  
terms for licensed third-party software  
components included within ICU  
libraries.

#### 1. ICU License - ICU 1.8.1 to ICU 57.1

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated documentation files (the  
"Software"), to deal in the Software without restriction, including  
without limitation the rights to use, copy, modify, merge, publish,  
distribute, and/or sell copies of the Software, and to permit persons  
to whom the Software is furnished to do so, provided that the above  
copyright notice(s) and this permission notice appear in all copies of  
the Software and that both the above copyright notice(s) and this  
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE

AND NONINFRINGEMENT  
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR  
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY  
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER  
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF  
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN  
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder  
shall not be used in advertising or otherwise to promote the sale, use  
or other dealings in this Software without prior written authorization  
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the  
property of their respective owners.

From: <https://www.unicode.org/copyright.html>:

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the [Unicode Privacy Policy](#).

Unicode Copyright

Copyright 1991-2020

Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

Unicode Software ("SOFTWARE") includes any source code published in the Unicode Standard  
or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any  
person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming

to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions (Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

#### Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

#### Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

#### Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

#### Trademarks & Logos

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy)

are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

#### Miscellaneous

**Jurisdiction and Venue.** This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

**Modification by Unicode, Inc.** Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

**Taxes.** The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

**Severability.** If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

**Entire Agreement.** This Agreement constitutes the entire agreement between the parties.

...

## Mesa 3-D Graphics Library v21.0.3

### Mesa License

...

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Attention, Contributors

When contributing to the Mesa project you must agree to the licensing terms of the component to which you're contributing.

The following section lists the primary components of the Mesa distribution and their respective licenses.

Mesa Component Licenses

Component	Location	License
Main Mesa code	src/mesa/	MIT
Device drivers	src/mesa/drivers/*	MIT, generally
Gallium code	src/gallium/	MIT
Ext headers	GL/gltext.h	Khronos
	GL/gltext.h	Khronos
	GL/wglext.h	Khronos
	KHR/khrplatform.h	Khronos

\*\*\*\*\*

----

include/GL/gl.h :

Mesa 3-D  
graphics library

Copyright (C) 1999-2006 Brian Paul All Rights Reserved.  
Copyright (C) 2009 VMware, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR

OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*

----

```
include/GL/glext.h
include/GL/glxt.h
include/GL/wglxt.h :
```

Copyright (c) 2013 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

\*\*\*\*\*

----

```
include/KHR/khrplatform.h :
```

Copyright (c) 2008 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do

so, subject to  
the following conditions:

The above copyright notice and this permission notice shall be included  
in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.  
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY  
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

\*\*\*\*\*

---

## Mozilla Public Suffix List

### Public Suffix Notice

---

You are receiving a copy of the Mozilla Public Suffix List in the following  
file: <java-home>/lib/security/public\_suffix\_list.dat. The terms of the  
Oracle license do NOT apply to this file; it is licensed under the  
Mozilla Public License 2.0, separately from the Oracle programs you receive.  
If you do not wish to use the Public Suffix List, you may remove the  
<java-home>/lib/security/public\_suffix\_list.dat file.

The Source Code of this file is available under the  
Mozilla Public License, v. 2.0 and is located at  
[https://raw.githubusercontent.com/publicsuffix/list/3c213aab32b3c014f171b1673d4ce9b5cd72bf1c/public\\_suffix\\_list.dat](https://raw.githubusercontent.com/publicsuffix/list/3c213aab32b3c014f171b1673d4ce9b5cd72bf1c/public_suffix_list.dat).

If a copy of the MPL was not distributed with this file, you can obtain one  
at <https://mozilla.org/MPL/2.0/>.

Software distributed under the License is distributed on an "AS IS" basis,  
WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License  
for the specific language governing rights and limitations  
under the License.

---

### MPL v2.0

---

Mozilla Public License Version 2.0

=====

1. Definitions

-----

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to,



deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

-----

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

-----

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms

of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3.

#### Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the

Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

-----

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must:

(a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

-----

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become

compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

\*\*\*\*\*

\*

\*

\* 6. Disclaimer of Warranty

\*

\* -----

\*

\*

\*

\* Covered Software is provided under this License on an "as is" \*  
\* basis, without warranty of any kind, either expressed, implied, or \*  
\* statutory, including, without limitation, warranties that the \*  
\* Covered Software is free of defects, merchantable, fit for a \*  
\* particular purpose or non-infringing. The entire risk as to the \*  
\* quality and performance of the Covered Software is with You. \*  
\* Should any Covered Software prove defective in any respect, You \*  
\* (not any Contributor) assume the cost of any necessary servicing, \*  
\* repair, or correction. This disclaimer of warranty constitutes an \*

\*

essential part of this License. No use of any Covered Software is \*  
\* authorized under this License except under this disclaimer. \*

\*

\*

\*\*\*\*\*

\*\*\*\*\*

\*

\*

\* 7. Limitation of Liability \*

\* ----- \*

\* \*

\* Under no circumstances and under no legal theory, whether tort \*

\* (including negligence), contract, or otherwise, shall any \*

\* Contributor, or anyone who distributes Covered Software as \*

\* permitted above, be liable to You for any direct, indirect, \*

\* special, incidental, or consequential damages of any \*

character \*

\* including, without limitation, damages for lost profits, loss of \*

\* goodwill, work stoppage, computer failure or malfunction, or any \*

\* and all other commercial damages or losses, even if such party \*

\* shall have been informed of the possibility of such damages. This \*

\* limitation of liability shall not apply to liability for death or \*

\* personal injury resulting from such party's negligence to the \*

\* extent applicable law prohibits such limitation. Some \*

\* jurisdictions do not allow the exclusion or limitation of \*

\* incidental or consequential damages, so this exclusion and \*

\* limitation may not apply to You. \*

\* \*

\*\*\*\*\*

## 8. Litigation

-----

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

-----

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

-----

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License.

Each version will be given a distinguishing version number.

#### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

#### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

#### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

##### Exhibit A - Source Code Form License Notice

-----

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

##### Exhibit B - "Incompatible With Secondary Licenses" Notice

-----

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

...

Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.  
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.

This code is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 only, as published by the Free Software Foundation.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License version 2 for more details (a copy is included in the LICENSE file that accompanied this code).

You should have received a copy of the GNU General Public License version 2 along with this work; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA or visit [www.oracle.com](http://www.oracle.com) if you need additional information or have any questions.

## Bert Belder: wepoll v 1.5.8

### wepoll License

...

wepoll - epoll for Windows  
<https://github.com/piscisaureus/wepoll>

Copyright 2012-2020, Bert Belder <bertbelder@gmail.com>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR



ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

...

#### UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

#### COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2021 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,

DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

# 1.170 lz4-java 1.8.0

## 1.170.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.171 ca-certificates 20230506-r0

## 1.171.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
* You may opt to use, copy, modify, merge, publish, distribute and/or sell
* furnished to do so, under the terms of the COPYING file.
```

Found in path(s):

```
* /opt/cola/permits/1667253608_1683577718.267003/0/ca-certificates-20230506-tar-bz2/ca-certificates-
20230506/mk-ca-bundle.pl
```

No license file was found, but licenses were detected in source scan.

```
/* c_rehash.c - Create hash symlinks for certificates
```

\* C implementation based on the original Perl and shell versions  
\*  
\* Copyright (c) 2013-2014 Timo Teräs <timo.teras@iki.fi>  
\* All rights reserved.  
\*  
\* This software is licensed under the MIT License.  
\* Full license available at: <http://opensource.org/licenses/MIT>  
\*/

Found in path(s):

\* /opt/cola/permits/1667253608\_1683577718.267003/0/ca-certificates-20230506-tar-bz2/ca-certificates-20230506/c\_rehash.c

# 1.172 apk-tools 2.14.0-r2

## 1.172.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to

deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.



1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise

the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program

is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.  
END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.173 alpine-baselayout 3.4.3-r1

### 1.173.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Create mount directories in fstab
 *
 * Copyright(c) 2008 Natanael Copa <natanael.copa@gmail.com>
 * May be distributed under the terms of GPL-2
 *
 * usage: mkmntdirs [fstab]
 *
 */
```

Found in path(s):

```
* /opt/cola/permits/1675356458_1684887865.7744117/0/alpine-baselayout-master-zip/alpine-baselayout-master/src/mkmntdirs.c
```

## 1.174 ca-certificates-bundle 20230506-r0

### 1.174.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
* You may opt to use, copy, modify, merge, publish, distribute and/or sell
* furnished to do so, under the terms of the COPYING file.
```

Found in path(s):

\* /opt/cola/permits/1679450782\_1685340861.071418/0/ca-certificates-20230506-tar-bz2/ca-certificates-20230506/mk-ca-bundle.pl

No license file was found, but licenses were detected in source scan.

/\* c\_rehash.c - Create hash symlinks for certificates

\* C implementation based on the original Perl and shell versions

\*

\* Copyright (c) 2013-2014 Timo Teräs <timo.teras@iki.fi>

\* All rights reserved.

\*

\* This software is licensed under the MIT License.

\* Full license available at: <http://opensource.org/licenses/MIT>

\*/

Found in path(s):

\* /opt/cola/permits/1679450782\_1685340861.071418/0/ca-certificates-20230506-tar-bz2/ca-certificates-20230506/c\_rehash.c

## 1.175 libc-utils 0.7.2-r5

### 1.175.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 1991, 1993

\* The Regents of the University of California. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND

\* ANY EXPRESS OR IMPLIED WARRANTIES,

INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.  
\*  
\* @(#)queue.h 8.5 (Berkeley) 8/20/94  
\*/

Found in path(s):

\* /opt/cola/permits/1686513755\_1685082238.691036/0/aports-master-main-libc-dev-  
zip/aports-master-main-libc-dev/main/libc-dev/sys-queue.h

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright 2002 Niels Provos <provos@citi.umich.edu>

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

\* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

\* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

\* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT

\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

\* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

\* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

\* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

\* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

Found in path(s):

\* /opt/cola/permits/1686513755\_1685082238.691036/0/aports-master-main-libc-dev-  
zip/aports-master-main-libc-dev/main/libc-dev/sys-tree.h

## 1.176 jackson-annotations 2.15.1



## 1.176.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

## 1.177 jackson-databind 2.15.1

### 1.177.1 Available under license :

Apache License  
Version 2.0, January 2004

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the



origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.

It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0

To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

## 1.178 jackson 2.15.1

### 1.178.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.  
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has  
been in development since 2007.  
It is currently developed by a community of developers.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0  
To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included  
in some artifacts (usually source distributions); but is always available  
from the source code management (SCM) system project uses.

## FastDoubleParser

jackson-core bundles a shaded copy of FastDoubleParser <<https://github.com/wrandelshofer/FastDoubleParser>>.  
That code is available under an MIT license  
<<https://github.com/wrandelshofer/FastDoubleParser/blob/main/LICENSE>>  
under the following  
copyright.

Copyright 2023 Werner Randelshofer, Switzerland. MIT License.

See FastDoubleParser-NOTICE for details of other source code included in FastDoubleParser  
and the licenses and copyrights that apply to that code.

# 1.179 jackson-dataformat-yaml 2.15.1

## 1.179.1 Available under license :

This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License").  
See the License for details about distribution rights, and the specific rights regarding derivative works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.  
It is currently developed by a community of developers.

### ## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

### ## Licensing

Jackson components are licensed under Apache (Software) License, version 2.0, as per accompanying LICENSE file.

### ## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

# 1.180 alpine-baselayout-data 3.4.3-r1

## 1.180.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Create mount directories in fstab
 *
 * Copyright(c) 2008 Natanael Copa <natanael.copa@gmail.com>
 * May be distributed under the terms of GPL-2
 *
 * usage: mkmntdirs [fstab]
 *
```



\*/

Found in path(s):

\* /opt/cola/permits/1690300026\_1684996389.3603017/0/alpine-baselayout-master-zip/alpine-baselayout-master/src/mkmdir.c

## 1.181 opentelemetry-sdk 1.24.0

### 1.181.1 Available under license :

Apache-2.0

## 1.182 opentelemetry-semconv 1.24.0-alpha

### 1.182.1 Available under license :

Apache-2.0

## 1.183 opentelemetry-sdk-extension-autoconfigure-spi 1.24.0

### 1.183.1 Available under license :

Apache-2.0

## 1.184 scanelf 1.3.7-r1

### 1.184.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed

under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may



consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# 1.185 busybox 1.36.1

## 1.185.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source of bzip2 version 1.0.4. bzip2 source is distributed under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

-----  
GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for

a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution

of derivative or  
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity

of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA



Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of
Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.186 musl 1.2.4-r1

### 1.186.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

-----  
Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
Authors/contributors include:

A. Wilcox  
Ada Worcester  
Alex Dowad  
Alex Suykov  
Alexander Monakov  
Andre McCurdy  
Andrew Kelley  
Anthony G. Basile  
Aric Belsito  
Arvid Picciani  
Bartosz Brachaczek  
Benjamin Peterson  
Bobby Bingham  
Boris Brezillon  
Brent Cook  
Chris Spiegel  
Clment Vasseur  
Daniel Micay  
Daniel Sabogal  
Daurnimator  
David Carlier  
David Edelsohn  
Denys Vlasenko  
Dmitry Ivanov  
Dmitry V. Levin  
Drew DeVault  
Emil Renner Berthing  
Fangrui Song  
Felix Fietkau  
Felix Janda

Gianluca Anzolin  
Hauke Mehrrens  
He X  
Hiltjo Posthuma  
Isaac Dunham  
Jaydeep Patil  
Jens Gustedt  
Jeremy Huntwork  
Jo-Philipp Wich  
Joakim Sindholt  
John Spencer  
Julien Ramseier  
Justin Cormack  
Kaarle Ritvanen  
Khem Raj  
Kylie McClain  
Leah Neukirchen  
Luca Barbato  
Luka  
Perkov  
M Farkas-Dyck (Strake)  
Mahesh Bodapati  
Markus Wichmann  
Masanori Oginō  
Michael Clark  
Michael Forney  
Mikhail Kremnyov  
Natanael Copa  
Nicholas J. Kain  
orc  
Pascal Cuoq  
Patrick Oppenlander  
Petr Hosek  
Petr Skocik  
Pierre Carrier  
Reini Urban  
Rich Felker  
Richard Pennington  
Ryan Fairfax  
Samuel Holland  
Segev Finer  
Shiz  
sin  
Solar Designer  
Stefan Kristiansson  
Stefan O'Rear  
Szabolcs Nagy  
Timo Ters

Trutz Behn  
Valentin Ochs  
Will Dietz  
William Haddon  
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (`src/string/aarch64/*`) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (`src/crypt/crypt_des.c`) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/crypt/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsort.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86\_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the elcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/\* and arch/\*/bits/\*) and crt files intended to be linked into applications (crt/\*, ldso/dlstart.c, and arch/\*/crt\_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham  
John Spencer  
Nicholas J. Kain  
Rich Felker  
Richard Pennington  
Stefan Kristiansson  
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

# 1.187 reload4j 2.20.0

## 1.187.1 Available under license :

Apache Log4j Core

Copyright 1999-2012 Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java

Copyright 2005-2006 Tim Fennell

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions



for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Log4j

Copyright 1999-2012 Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Dumbster SMTP test server

Copyright 2004 Jason Paul Kitchen

Apache Log4j

Copyright 1999-2023 Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java

Copyright 2005-2006 Tim Fennell

Dumbster SMTP test server  
Copyright 2004 Jason Paul Kitchen

TypeUtil.java  
Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

picocli (<http://picocli.info>)  
Copyright 2017 Remko Popma

TimeoutBlockingWaitStrategy.java and parts of Util.java  
Copyright 2011 LMAX Ltd.

## 1.188 slf4j 2.0.7

### 1.188.1 Available under license :

Copyright (c) 2004-2023 QOS.ch  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.189 snake-yaml 2.6

### 1.189.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.190 kotlin-reflect 1.8.10

### 1.190.1 Available under license :

Apache-2.0

## 1.191 kotlin 1.8.10

## 1.191.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
{ "version":3,"file":"kotlin.js","sources":["wrapper.js","js/arrayUtils.js","js/callableReferenceUtils.js","js/conversions.js","js/core.js","js/long.js","js/markerFunctions.js","js/misc.js","js/polyfills.js","js/rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","common/src/generated/_Arrays.kt","common/src/generated/_Ranges.kt","unsigned/src/kotlin/UByte.kt","unsigned/src/kotlin/UInt.kt","unsigned/src/kotlin/UShort.kt","src/kotlin/collections/Collections.kt","src/kotlin/collections/Maps.kt","src/kotlin/collections/Sets.kt","src/kotlin/ranges/PrimitiveRanges.kt","src/kotlin/text/StringNumberConversions.kt","src/kotlin/time/Duration.kt","unsigned/src/kotlin/UnsignedUtils.kt","../core/builtins/src/kotlin/internal/InternalAnnotations.kt","src/kotlin/collections/Iterables.kt","src/kotlin/collections/Sequences.kt","src/kotlin/util/Preconditions.kt","js/src/generated/_ArraysJs.kt","src/kotlin/comparisons/Comparisons.kt","src/kotlin/util/Standard.kt","js/src/generated/_ComparisonsJs.kt","unsigned/src/kotlin/ULong.kt","common/src/generated/_Collections.kt","js/src/kotlin/collections.kt","src/kotlin/collections/Iterators.kt","common/src/generated/_Comparisons.kt","common/src/generated/_Maps.kt","common/src/generated/_OneToManyTitlecaseMappings.kt","js/src/kotlin/text/char.kt","js/src/kotlin/text/string.kt","src/kotlin/text/Char.kt","src/kotlin/CharCode.kt","common/src/generated/_Sequences.kt","common/src/generated/_Sets.kt","common/src/generated/_Strings.kt","src/kotlin/text/Strings.kt","unsigned/src/kotlin/UByteArray.kt","unsigned/src/kotlin/UIntArray.kt","unsigned/src/kotlin/ULongArray.kt","unsigned/src/kotlin/UShortArray.kt","common/src/generated/_UArrays.kt","common/src/generated/_UCollections.kt","common/src/generated/_UComparisons.kt","common/src/generated/_URanges.kt","common/src/generated/_USequences.kt","common/src/kotlin/ExceptionsH.kt","common/src/kotlin/JsAnnotationsH.kt","common/src/kotlin/ioH.kt","builtin-sources/Collections.kt","builtin-sources/Unit.kt","builtin-sources/annotation/Annotations.kt","src/kotlin/builtins.kt","src/kotlin/jsTypeOf.kt","src/kotlin/kotlin.kt","src/kotlin/CharCode_js-v1.kt","src/kotlin/coroutines/CoroutineImpl.kt","src/kotlin/util/Result.kt","src/kotlin/coroutines/Continuation.kt","src/kotlin/coroutines/intrinsics/IntrinsicsJs.kt","src/kotlin/currentBeMisc.kt","src/kotlin/exceptions.kt","src/kotlin/jsOperators.kt","src/kotlin/math_js-v1.kt","src/kotlin/numbers_js-v1.kt","src/kotlin/reflection_js-v1.kt","src/kotlin/text/numberConversions_js-v1.kt","js/src/kotlin/js.arrays/fill.kt","js/src/kotlin/js.arrays/sort.kt","js/src/generated/_CharCategories.kt","js/src/generated/_CollectionsJs.kt","js/src/generated/_DigitChars.kt","js/src/generated/_LetterChars.kt","js/src/generated/_OtherLowercaseChars.kt","js/src/generated/_OtherUppercaseChars.kt","js/src/generated/_StringsJs.kt","js/src/generated/_TitlecaseMappings.kt","js/src/generated/_UArraysJs.kt","js/src/generated/_WhitespaceChars.kt","js/src/kotlin/Comparator.kt","js/src/kotlin/annotations.kt","js/src/kotlin/annotationsJVM.kt","js/src/kotlin/collections/AbstractMutableCollection.kt","js/src/kotlin/collections/AbstractMutableList.kt","js/src/kotlin/collections/AbstractMutableMap.kt","js/src/kotlin/collections/AbstractMutableSet.kt","js/src/kotlin/collections/ArrayList.kt","js/src/kotlin/collections/ArraySorting.kt","js/src/kotlin/collections/ArraysJs.kt","js/src/kotlin/collections/EqualityComparator.kt","js/src/kotlin/collections/HashMap.kt","js/src/kotlin/collections/HashSet.kt","js/src/kotlin/collections/InternalHashCodeMap.kt","js/src/kotlin/collections/InternalMap.kt","js/src/kotlin/collections/InternalStringMap.kt","js/src/kotlin/collections/LinkedHashMap.kt","js/src/kotlin/collections/LinkedHashSet.kt","js/src/kotlin/concurrent.kt","js/src/kotlin/console.kt","js/src/kotlin/coroutines/SafeContinuationJs.kt","js/src/kotlin/coroutines/cancellation/CancellationException.kt","js/src/kotlin/coroutines/js/internal/EmptyContinuation.kt","js/src/kotlin/date.kt","js/src/kotlin/dom/Builders.kt","js/src/kotlin/dom/Classes.kt","js/src/kotlin/dom/Dom.kt","js/src/kotlin/dom/EventListener.kt","js/src/kotlin/dom/ItemArrayLike.kt","js/src/kotlin/dom/Mutations.kt","js/src/kotlin/dynamic.kt","js/src/kotlin/enums/EnumEntriesSerializationProxy.kt","js/src/kotlin/exceptionUtils.kt","js/src/kotlin/grouping.kt","src/kotlin/collections/Grouping.kt","js/src/kotlin/internalAnnotations.kt","js/src/kotlin/json.kt","js/src/kotlin/math.kt","js/src/kotlin/numbers.kt","js/src/kotlin/promise.kt","js/src/kotlin/random/PlatformRandom.kt","js/src/kotlin/reflect/AssociatedObjects.kt","js/src/kotlin/reflect/JsClass.kt","js/src/kotlin/reflect/KClassImpl.kt","js/src/kotlin/reflect/KClassesImpl.kt","js/src/kotlin/reflect/KTypeHelpers.kt","js/src/kotlin/reflect/KTypeImpl.kt","js/src/kotlin/reflect/KTypeParameterImpl.kt","js/src/kotlin/reflect/primitives.kt","js/src/kotlin/reflect/reflection.kt","js/src/kotlin/regexp.kt","js/src/kotlin/sequence.kt","js/src/kotlin/text/Ch
```



arCategoryJS.kt", "js/src/kotlin/text/CharacterCodingExceptionJs.kt", "js/src/kotlin/text/StringBuilderJs.kt", "js/src/kotlin/text/numberConversions.kt", "js/src/kotlin/text/regex.kt", "src/kotlin/text/StringBuilder.kt", "js/src/kotlin/text/stringCode.kt", "js/src/kotlin/text/utf8Encoding.kt", "js/src/kotlin/throwableExtensions.kt", "js/src/kotlin/time/DurationJs.kt", "js/src/kotlin/time/DurationUnit.kt", "js/src/kotlin/time/MonoTimeSource.kt", "js/src/kotlinx/dom/Builders.kt", "js/src/kotlinx/dom/Classes.kt", "src/kotlin/text/regex/RegexExtensions.kt", "js/src/kotlinx/dom/Dom.kt", "js/src/kotlinx/dom/Mutations.kt", "js/src/org.w3c/deprecated.kt", "js/src/org.w3c/org.khronos.webgl.kt", "js/src/org.w3c/org.w3c.dom.clipboard.kt", "js/src/org.w3c/org.w3c.dom.css.kt", "js/src/org.w3c/org.w3c.dom.encryptedmedia.kt", "js/src/org.w3c/org.w3c.dom.events.kt", "js/src/org.w3c/org.w3c.dom.kt", "js/src/org.w3c/org.w3c.fetch.kt", "js/src/org.w3c/org.w3c.dom.mediacapture.kt", "js/src/org.w3c/org.w3c.dom.mediasource.kt", "js/src/org.w3c/org.w3c.dom.pointerevents.kt", "js/src/org.w3c/org.w3c.dom.svg.kt", "js/src/org.w3c/org.w3c.files.kt", "js/src/org.w3c/org.w3c.notifications.kt", "js/src/org.w3c/org.w3c.workers.kt", "js/src/org.w3c/org.w3c.xhr.kt", "src/kotlin/annotations/ExperimentalStdlibApi.kt", "src/kotlin/annotations/Inference.kt", "src/kotlin/annotations/Multiplatform.kt", "src/kotlin/annotations/OptIn.kt", "src/kotlin/annotations/WasExperimental.kt", "src/kotlin/collections/AbstractCollection.kt", "src/kotlin/collections/AbstractIterator.kt", "src/kotlin/collections/AbstractList.kt", "src/kotlin/collections/AbstractMap.kt", "src/kotlin/collections/AbstractSet.kt", "src/kotlin/collections/ArrayDeque.kt", "src/kotlin/collections/Arrays.kt", "src/kotlin/collections/IndexedValue.kt", "src/kotlin/collections/MapAccessors.kt", "src/kotlin/collections/MapWithDefault.kt", "src/kotlin/collections/MutableCollections.kt", "src/kotlin/collections/PrimitiveIterators.kt", "src/kotlin/collections/ReversedViews.kt", "src/kotlin/collections/SequenceBuilder.kt", "src/kotlin/collections/SlidingWindow.kt", "src/kotlin/collections/UArraySorting.kt", "src/kotlin/comparisons/compareTo.kt", "src/kotlin/contracts/ContractBuilder.kt", "src/kotlin/coroutines/ContinuationInterceptor.kt", "src/kotlin/coroutines/CoroutineContext.kt", "src/kotlin/coroutines/CoroutineContextImpl.kt", "src/kotlin/coroutines/intrinsics/Intrinsics.kt", "src/kotlin/enums/EnumEntries.kt", "src/kotlin/experimental/ExperimentalObjCName.kt", "src/kotlin/experimental/ExperimentalObjCRefinement.kt", "src/kotlin/experimental/bitwiseOperations.kt", "src/kotlin/experimental/inferenceMarker.kt", "src/kotlin/internal/Annotations.kt", "src/kotlin/internal/progressionUtil.kt", "src/kotlin/properties/Delegates.kt", "src/kotlin/properties/Interfaces.kt", "src/kotlin/properties/ObservableProperty.kt", "src/kotlin/properties/PropertyReferenceDelegates.kt", "src/kotlin/random/Random.kt", "src/kotlin/random/URandom.kt", "src/kotlin/random/XorWowRandom.kt", "src/kotlin/ranges/ProgressionIterators.kt", "src/kotlin/ranges/Progressions.kt", "src/kotlin/ranges/Range.kt", "src/kotlin/ranges/Ranges.kt", "src/kotlin/reflect/KClasses.kt", "src/kotlin/reflect/KTypeProjection.kt", "src/kotlin/reflect/KVariance.kt", "src/kotlin/reflect/typeOf.kt", "src/kotlin/text/Appendable.kt", "src/kotlin/text/Indent.kt", "src/kotlin/text/Typography.kt", "src/kotlin/text/regex/MatchResult.kt", "src/kotlin/time/DurationUnit.kt", "src/kotlin/time/ExperimentalTime.kt", "src/kotlin/time/TimeSource.kt", "src/kotlin/time/TimeSources.kt", "src/kotlin/time/longSaturatedMath.kt", "src/kotlin/time/measureTime.kt", "src/kotlin/util/DeepRecursive.kt", "src/kotlin/util/FloorDivMod.kt", "src/kotlin/util/HashCode.kt", "src/kotlin/util/KotlinVersion.kt", "src/kotlin/util/Lateinit.kt", "src/kotlin/util/Lazy.kt", "src/kotlin/util/Numbers.kt", "src/kotlin/util/Suspend.kt", "src/kotlin/util/Tuples.kt", "unsigned/src/kotlin/UIntRange.kt", "unsigned/src/kotlin/ULongRange.kt", "unsigned/src/kotlin/UMath.kt", "unsigned/src/kotlin/UNumbers.kt", "unsigned/src/kotlin/UProgressionUtil.kt", "unsigned/src/kotlin/UStrings.kt", "unsigned/src/kotlin/annotations/Unsigned.kt", "common/src/kotlin/MathH.kt", "js/src/kotlin/js/js.math.kt"], "sourcesContent": ["(function (root, factory) {\n if (typeof define === 'function' && define.amd) {\n define('kotlin', ['exports'], factory);\n }\n else if (typeof exports === 'object') {\n factory(module.exports);\n }\n else {\n root.kotlin = {};\n factory(root.kotlin);\n }\n})(this, function (Kotlin) {\n var \_ = Kotlin;\n\n insertContent();\n});\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n \*\n\n Kotlin.isBooleanArray = function (a) {\n return (Array.isArray(a) || a instanceof Int8Array) && a.\$type\$ === \"BooleanArray\";\n};\n\n Kotlin.isByteArray = function (a) {\n return a instanceof Int8Array && a.\$type\$ !== \"BooleanArray\";\n};\n\n Kotlin.isShortArray = function (a) {\n return a instanceof Int16Array;\n};\n\n Kotlin.isCharArray = function (a) {\n return a instanceof Uint16Array && a.\$type\$ === \"CharArray\";\n};\n\n Kotlin.isIntArray = function (a) {\n return a instanceof Int32Array;\n};\n\n Kotlin.isFloatArray = function (a) {\n return a instanceof

```

Float32Array\n};\n\nKotlin.isDoubleArray = function (a) {\n return a instanceof
Float64Array\n};\n\nKotlin.isLongArray = function (a) {\n return Array.isArray(a) && a.$type$ ===
\"LongArray\"\n};\n\nKotlin.isArray = function (a) {\n return Array.isArray(a) &&
!a.$type$;\n};\n\nKotlin.isArrayish = function (a) {\n return Array.isArray(a) ||
ArrayBuffer.isView(a)\n};\n\nKotlin.arrayToString = function (a) {\n if (a === null) return \"null\"\n var
toString = Kotlin.isCharArray(a) ? String.fromCharCode : Kotlin.toString;\n
 return \"[\" + Array.prototype.map.call(a, function(e) { return toString(e); }).join(\", \") +
\"]\";\n};\n\nKotlin.arrayDeepToString = function (arr) {\n return
Kotlin.kotlin.collections.contentDeepToStringImpl(arr);\n};\n\nKotlin.arrayEquals = function (a, b) {\n if (a ===
b) {\n return true;\n }\n if (a === null || b === null || !Kotlin.isArrayish(b) || a.length !== b.length) {\n
return false;\n }\n for (var i = 0, n = a.length; i < n; i++) {\n if (!Kotlin.equals(a[i], b[i])) {\n return
false;\n }\n }\n return true;\n};\n\nKotlin.arrayDeepEquals = function (a, b) {\n return
Kotlin.kotlin.collections.contentDeepEqualsImpl(a, b);\n};\n\nKotlin.arrayHashCode = function (arr) {\n if (arr
=== null) return 0\n var result = 1;\n for (var i = 0, n = arr.length; i < n; i++) {\n result = ((31 * result | 0) +
Kotlin.hashCode(arr[i])) | 0;\n }\n return result;\n};\n\nKotlin.arrayDeepHashCode
= function (arr) {\n return
Kotlin.kotlin.collections.contentDeepHashCodeImpl(arr);\n};\n\nKotlin.primitiveArraySort = function (array) {\n
array.sort(Kotlin.doubleCompareTo)\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.getCallableRef = function(name, f) {\n f.callableName = name;\n return
f;\n};\n\nKotlin.getPropertyCallableRef = function(name, paramCount, getter, setter) {\n getter.get = getter;\n
getter.set = setter;\n getter.callableName = name;\n return getPropertyRefClass(getter, setter,
propertyRefClassMetadataCache[paramCount]);\n};\n\nfunction getPropertyRefClass(obj, setter, cache) {\n
obj.$metadata$ = getPropertyRefMetadata(typeof setter === \"function\" ? cache.mutable : cache.immutable);\n
obj.constructor = obj;\n return obj;\n}\n\nvar propertyRefClassMetadataCache
= [\n {\n mutable: { value: null, implementedInterface: function () {\n return
Kotlin.kotlin.reflect.KMutableProperty0 }\n },\n immutable: { value: null, implementedInterface: function
() {\n return Kotlin.kotlin.reflect.KProperty0 }\n }\n },\n {\n mutable: { value: null,
implementedInterface: function () {\n return Kotlin.kotlin.reflect.KMutableProperty1 }\n },\n
immutable: { value: null, implementedInterface: function () {\n return Kotlin.kotlin.reflect.KProperty1 }\n
}\n }\n];\n\nfunction getPropertyRefMetadata(cache) {\n if (cache.value === null) {\n cache.value = {\n
interfaces: [cache.implementedInterface()],\n baseClass: null,\n functions: {},\n properties:
{},\n types: {},\n staticMembers: {} }\n }\n return cache.value;\n}\n\n\"/*\n * Copyright
2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.toShort = function (a) {\n return (a &
0xFFFF) << 16 >> 16;\n};\n\nKotlin.toByte = function (a) {\n return (a & 0xFF) << 24 >>
24;\n};\n\nKotlin.toChar = function (a) {\n return a & 0xFFFF;\n};\n\nKotlin.numberToLong = function (a) {\n
return a instanceof Kotlin.Long ? a : Kotlin.Long.fromNumber(a);\n};\n\nKotlin.numberToInt = function (a) {\n
return a instanceof Kotlin.Long ? a.toInt() : Kotlin.doubleToInt(a);\n};\n\nKotlin.numberToShort = function (a) {\n
return Kotlin.toShort(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToByte = function (a) {\n return
Kotlin.toByte(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToDouble = function (a) {\n return
+a;\n};\n\nKotlin.numberToChar = function (a) {\n return
Kotlin.toChar(Kotlin.numberToInt(a));\n};\n\nKotlin.doubleToInt = function(a)
{\n if (a > 2147483647) return 2147483647;\n if (a < -2147483648) return -2147483648;\n return a |
0;\n};\n\nKotlin.toBoxedChar = function (a) {\n if (a == null) return a;\n if (a instanceof Kotlin.BoxedChar)
return a;\n return new Kotlin.BoxedChar(a);\n};\n\nKotlin.unboxChar = function(a) {\n if (a == null) return a;\n
return Kotlin.toChar(a);\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language

```

```

contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.equals = function (obj1, obj2) {\n if (obj1 == null) {\n return obj2 ==
null;\n }\n if (obj2 == null) {\n return false;\n }\n if (obj1 !== obj2) {\n return obj2 !== obj2;\n
}\n if (typeof obj1 === "object" && typeof obj1.equals === "function") {\n return obj1.equals(obj2);\n
}\n if (typeof obj1 === "number" && typeof obj2 ===
"number") {\n return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2)\n }\n return obj1 ===
obj2;\n};\n\nKotlin.hashCode = function (obj) {\n if (obj == null) {\n return 0;\n }\n var objType = typeof
obj;\n if ("object" === objType) {\n return "function" === typeof obj.hashCode ? obj.hashCode() :
getObjectHashCode(obj);\n }\n if ("function" === objType) {\n return getObjectHashCode(obj);\n }\n
if ("number" === objType) {\n return Kotlin.numberHashCode(obj);\n }\n if ("boolean" === objType)
{\n return Number(obj)\n }\n var str = String(obj);\n return
getStringHashCode(str);\n};\n\nKotlin.toString = function (o) {\n if (o == null) {\n return "null";\n }\n
else if (Kotlin.isArrayish(o)) {\n return "["+o+"]";\n }\n else {\n return o.toString();\n }\n};\n\n/**
 * @const *\nvar POW_2_32 = 4294967296;\n// TODO: consider switching to Symbol type
once we are on ES6.\n/** @const *\nvar OBJECT_HASH_CODE_PROPERTY_NAME =
"kotlinHashCodeValue$";\n\nfunction getObjectHashCode(obj) {\n if
(! (OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {\n var hash = (Math.random() * POW_2_32) | 0; //
Make 32-bit signed integer.\n Object.defineProperty(obj, OBJECT_HASH_CODE_PROPERTY_NAME, {\n
value: hash, enumerable: false });\n }\n return
obj[OBJECT_HASH_CODE_PROPERTY_NAME];\n}\n\nfunction getStringHashCode(str) {\n var hash = 0;\n
for (var i = 0; i < str.length; i++) {\n var code = str.charCodeAt(i);\n hash = (hash * 31 + code) | 0; // Keep
it 32-bit.\n }\n return hash;\n}\n\nKotlin.identityHashCode = getObjectHashCode;\n"/**\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by
the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Copyright 2009 The Closure
Library Authors. All Rights Reserved.\n\n//
Licensed under the Apache License, Version 2.0 (the "License");\n// you may not use this file except in
compliance with the License.\n// You may obtain a copy of the License at\n\n//
http://www.apache.org/licenses/LICENSE-2.0\n\n// Unless required by applicable law or agreed to in writing,
software\n// distributed under the License is distributed on an "AS-IS" BASIS,\n// WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied.\n\n/**\n * Constructs a 64-bit two's-complement integer,
given its low and high 32-bit\n * values as *signed* integers. See the from* functions below for more\n *
convenient ways of constructing Longs.\n * \n * The internal representation of a long is the two given signed, 32-bit
values.\n * We use 32-bit pieces because these are the size of integers on which\n * Javascript performs bit-
operations. For operations like addition and\n * multiplication, we split each number into 16-bit pieces, which can
easily be\n * multiplied
within Javascript's floating-point representation without overflow\n * or change in sign.\n * \n * In the algorithms
below, we frequently reduce the negative case to the\n * positive case by negating the input(s) and then post-
processing the result.\n * Note that we must ALWAYS check specially whether those values are MIN_VALUE\n *
(-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as\n * a positive number, it
overflows back into a negative). Not handling this\n * case would often result in infinite recursion.\n * \n * @param
{number} low The low (signed) 32 bits of the long.\n * @param {number} high The high (signed) 32 bits of the
long.\n * @constructor\n * @final\n */\n\nKotlin.Long = function(low, high) {\n /**\n * @type {number}\n *
@private\n */\n this.low_ = low | 0; // force into 32 signed bits.\n\n /**\n * @type {number}\n *
@private\n */\n this.high_ = high | 0; // force into 32 signed bits.\n};\n\nKotlin.Long.$metadata$ = {\n kind:
"class",\n simpleName: "Long",\n interfaces: []\n};\n\n// NOTE: Common constant values ZERO, ONE,
NEG_ONE, etc. are defined below the\n// from* methods on which they depend.\n\n/**\n * A cache of the Long
representations of small integer values.\n * @type {!Object}\n * @private\n */\n\nKotlin.Long.IntCache_ =
{};\n\n/**\n * Returns a Long representing the given (32-bit) integer value.\n * @param {number} value The 32-

```

```

bit integer in question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromInt =
function(value) {\n if (-128 <= value && value < 128) {\n var cachedObj = Kotlin.Long.IntCache_[value];\n if
(cachedObj) {\n return cachedObj;\n }\n }\n\n var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);\n if (-
128 <= value && value < 128) {\n Kotlin.Long.IntCache_[value] = obj;\n }\n return obj;\n};\n\n\n/**\n *
Converts this number value to `Long`.\n * The fractional part, if any, is rounded down towards zero.\n * Returns
zero if this `Double` value is `NaN`, `Long.MIN_VALUE` if it's less than `Long.MIN_VALUE`,\n *
`Long.MAX_VALUE` if it's bigger than `Long.MAX_VALUE`.\n * @param {number} value The number in
question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromNumber =
function(value) {\n if (isNaN(value)) {\n return Kotlin.Long.ZERO;\n } else if (value <= -
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MIN_VALUE;\n } else if (value + 1 >=
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MAX_VALUE;\n } else if (value < 0) {\n return
Kotlin.Long.fromNumber(-value).negate();\n } else {\n return new Kotlin.Long(\n (value %
Kotlin.Long.TWO_PWR_32_DBL_) | 0,\n (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0);\n
}\n};\n\n\n/**\n * Returns a Long representing the 64-bit integer that comes by concatenating\n * the given high and
low bits. Each is assumed to use 32 bits.\n * @param {number} lowBits The low 32-bits.\n * @param {number}
highBits The high 32-bits.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromBits
= function(lowBits, highBits) {\n return new Kotlin.Long(lowBits, highBits);\n};\n\n\n/**\n * Returns a Long
representation of the given string, written using the given\n * radix.\n * @param {string} str The textual
representation of the Long.\n * @param {number=} opt_radix The radix in which the text is written.\n * @return
{!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromString = function(str, opt_radix) {\n if
(str.length == 0) {\n throw Error('number format error: empty string');\n }\n\n var radix = opt_radix || 10;\n if
(radix < 2 || 36 < radix) {\n throw Error('radix out of range: ' + radix);\n }\n\n if (str.charAt(0) == '-') {\n return
Kotlin.Long.fromString(str.substring(1), radix).negate();\n } else if (str.indexOf('-') >= 0) {\n throw Error('number
format error: interior \"-\" character: ' + str);\n }\n\n // Do several (8) digits
each time through the loop, so as to\n // minimize the calls to the very expensive emulated div.\n var
radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));\n\n var result = Kotlin.Long.ZERO;\n for (var i =
0; i < str.length; i += 8) {\n var size = Math.min(8, str.length - i);\n var value = parseInt(str.substring(i, i + size),
radix);\n if (size < 8) {\n var power = Kotlin.Long.fromNumber(Math.pow(radix, size));\n result =
result.multiply(power).add(Kotlin.Long.fromNumber(value));\n } else {\n result =
result.multiply(radixToPower);\n result = result.add(Kotlin.Long.fromNumber(value));\n }\n }\n return
result;\n};\n\n\n// NOTE: the compiler should inline these constant values below and then remove\n// these
variables, so there should be no runtime penalty for these.\n\n\n/**\n * Number used repeated below in calculations.
This must appear before the\n * first call to any from* function below.\n * @type {number}\n * @private\n
*/\nKotlin.Long.TWO_PWR_16_DBL_
= 1 << 16;\n\n\n/**\n * @type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_24_DBL_ = 1 <<
24;\n\n\n/**\n * @type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_32_DBL_ =\n
Kotlin.Long.TWO_PWR_16_DBL_ * Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n *
@private\n */\nKotlin.Long.TWO_PWR_31_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ / 2;\n\n\n/**\n *
@type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_48_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_
* Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*/\nKotlin.Long.TWO_PWR_64_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_32_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*/\nKotlin.Long.TWO_PWR_63_DBL_ =\n Kotlin.Long.TWO_PWR_64_DBL_ / 2;\n\n\n/**\n * @type
{!Kotlin.Long} */\nKotlin.Long.ZERO = Kotlin.Long.fromInt(0);\n\n\n/**\n * @type {!Kotlin.Long}
*/\nKotlin.Long.ONE = Kotlin.Long.fromInt(1);\n\n\n/**\n * @type {!Kotlin.Long} */\nKotlin.Long.NEG_ONE =
Kotlin.Long.fromInt(-1);\n\n\n/**\n * @type {!Kotlin.Long} */\nKotlin.Long.MAX_VALUE =\n
Kotlin.Long.fromBits(0xFFFFFFFF | 0, 0x7FFFFFFF
| 0);\n\n\n/**\n * @type {!Kotlin.Long} */\nKotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 |

```



```

Long is greater than or equal to the other.\n *\nKotlin.Long.prototype.greaterThanOrEqual = function(other) {\n
return this.compare(other) >= 0;\n};\n\n/**\n * Compares this Long with the given one.\n * @param
{Kotlin.Long} other Long to compare against.\n * @return {number} 0 if they are the same, 1 if the this is greater,
and -1\n * if the given one is greater.\n *\nKotlin.Long.prototype.compare = function(other) {\n if
(this.equalsLong(other)) {\n return 0;\n }\n\n var thisNeg = this.isNegative();\n var otherNeg =
other.isNegative();\n if (thisNeg && !otherNeg) {\n return -1;\n }\n if (!thisNeg && otherNeg) {\n return 1;\n
}\n\n // at this point, the signs are the same, so subtraction will not overflow\n if (this.subtract(other).isNegative())
{\n return -1;\n } else {\n return 1;\n }\n};\n\n/**\n * @return {!Kotlin.Long} The negation of this value.\n
*\nKotlin.Long.prototype.negate = function() {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return
Kotlin.Long.MIN_VALUE;\n } else {\n return this.not().add(Kotlin.Long.ONE);\n }\n};\n\n/**\n * Returns the
sum of this and the given Long.\n * @param {Kotlin.Long} other Long to add to this one.\n * @return
{!Kotlin.Long} The sum of this and the given Long.\n *\nKotlin.Long.prototype.add = function(other) {\n //
Divide each number into 4 chunks of 16 bits, and then sum the chunks.\n\n var a48 = this.high_ >>> 16;\n var a32
= this.high_ & 0xFFFF;\n var a16 = this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n var b48 = other.high_
>>> 16;\n var b32 = other.high_ & 0xFFFF;\n var b16 = other.low_ >>> 16;\n var b00 = other.low_ &
0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00 += a00 + b00;\n c16 += c00 >>> 16;\n c00 &=
0xFFFF;\n\n c16 += a16 + b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 + b32;\n c48 += c32 >>> 16;\n c32
&= 0xFFFF;\n c48 += a48 + b48;\n c48 &= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 <<
16) | c32);\n};\n\n/**\n * Returns the difference of this and the given Long.\n * @param {Kotlin.Long} other
Long to subtract from this.\n * @return {!Kotlin.Long} The difference of this and the given Long.\n
*\nKotlin.Long.prototype.subtract = function(other) {\n return this.add(other.negate());\n};\n\n/**\n * Returns
the product of this and the given long.\n * @param {Kotlin.Long} other Long to multiply with this.\n * @return
{!Kotlin.Long} The product of this and the other.\n *\nKotlin.Long.prototype.multiply = function(other) {\n if
(this.isZero()) {\n return Kotlin.Long.ZERO;\n } else if (other.isZero()) {\n return Kotlin.Long.ZERO;\n }\n\n if
(this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return other.isOdd() ? Kotlin.Long.MIN_VALUE :
Kotlin.Long.ZERO;\n\n } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return this.isOdd() ? Kotlin.Long.MIN_VALUE :
Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n if (other.isNegative()) {\n return
this.negate().multiply(other.negate());\n } else {\n return this.negate().multiply(other).negate();\n }\n } else if
(other.isNegative()) {\n return this.multiply(other.negate()).negate();\n }\n\n // If both longs are small, use float
multiplication\n if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&\n
other.lessThan(Kotlin.Long.TWO_PWR_24_)) {\n return Kotlin.Long.fromNumber(this.toNumber() *
other.toNumber());\n }\n\n // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.\n // We can
skip products that would overflow.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 =
this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ &
0xFFFF;\n var b16 = other.low_
>>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00 += a00 * b00;\n c16
+= c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 * b00;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c16 += a00
* b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 * b00;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n
c32 += a16 * b16;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a00 * b32;\n c48 += c32 >>> 16;\n c32 &=
0xFFFF;\n c48 += a48 * b00 + a32 * b16 + a16 * b32 + a00 * b48;\n c48 &= 0xFFFF;\n return
Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n/**\n * Returns this Long divided by the given
one.\n * @param {Kotlin.Long} other Long by which to divide.\n * @return {!Kotlin.Long} This Long divided by
the given one.\n *\nKotlin.Long.prototype.div = function(other) {\n if (other.isZero()) {\n throw Error('division
by zero');\n } else if (this.isZero()) {\n return Kotlin.Long.ZERO;\n }\n\n if
(this.equalsLong(Kotlin.Long.MIN_VALUE))

```

```

 {\n if (other.equalsLong(Kotlin.Long.ONE) ||\n other.equalsLong(Kotlin.Long.NEG_ONE)) {\n return
Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE == MIN_VALUE\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ONE;\n } else {\n // At this point,
we have |other| >= 2, so |this/other| < |MIN_VALUE|.\n var halfThis = this.shiftRight(1);\n var approx =
halfThis.div(other).shiftLeft(1);\n if (approx.equalsLong(Kotlin.Long.ZERO)) {\n return other.isNegative() ?
Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;\n } else {\n var rem = this.subtract(other.multiply(approx));\n
var result = approx.add(rem.div(other));\n return result;\n }\n }\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n
if (other.isNegative()) {\n return this.negate().div(other.negate());\n } else {\n return
this.negate().div(other).negate();\n
}\n } else if (other.isNegative()) {\n return this.div(other.negate()).negate();\n }\n\n // Repeat the following
until the remainder is less than other: find a\n // floating-point that approximates remainder / other *from below*,
add this\n // into the result, and subtract it from the remainder. It is critical that\n // the approximate value is less
than or equal to the real value so that the\n // remainder never becomes negative.\n var res = Kotlin.Long.ZERO;\n
var rem = this;\n while (rem.greaterThanOrEqual(other)) {\n // Approximate the result of division. This may be a
little greater or\n // smaller than the actual value.\n var approx = Math.max(1, Math.floor(rem.toNumber() /
other.toNumber()));\n\n // We will tweak the approximate result by changing it in the 48-th digit or\n // the
smallest non-fractional digit, whichever is larger.\n var log2 = Math.ceil(Math.log(approx) / Math.LN2);\n var
delta = (log2 <= 48) ? 1
: Math.pow(2, log2 - 48);\n\n // Decrease the approximation until it is smaller than the remainder. Note\n // that
if it is too large, the product overflows and is negative.\n var approxRes = Kotlin.Long.fromNumber(approx);\n
var approxRem = approxRes.multiply(other);\n while (approxRem.isNegative() || approxRem.greaterThan(rem))
{\n approx -= delta;\n approxRes = Kotlin.Long.fromNumber(approx);\n approxRem =
approxRes.multiply(other);\n }\n\n // We know the answer can't be zero... and actually, zero would cause\n //
infinite recursion since we would make no progress.\n if (approxRes.isZero()) {\n approxRes =
Kotlin.Long.ONE;\n }\n\n res = res.add(approxRes);\n rem = rem.subtract(approxRem);\n }\n return
res;\n};\n\n\n/**\n * Returns this Long modulo the given one.\n * @param {Kotlin.Long} other Long by which to
mod.\n * @return {!Kotlin.Long} This Long modulo the given one.\n */\nKotlin.Long.prototype.modulo =
function(other)
{\n return this.subtract(this.div(other).multiply(other));\n};\n\n\n/** @return {!Kotlin.Long} The bitwise-NOT of
this value. */\nKotlin.Long.prototype.not = function() {\n return Kotlin.Long.fromBits(~this.low_,
~this.high_);\n};\n\n\n/**\n * Returns the bitwise-AND of this Long and the given one.\n * @param {Kotlin.Long}
other The Long with which to AND.\n * @return {!Kotlin.Long} The bitwise-AND of this and the other.\n
*/\nKotlin.Long.prototype.and = function(other) {\n return Kotlin.Long.fromBits(this.low_ & other.low_,\n
this.high_ & other.high_);\n};\n\n\n/**\n * Returns the bitwise-OR of this Long and the given one.\n *
@param {Kotlin.Long} other The Long with which to OR.\n * @return {!Kotlin.Long} The bitwise-OR of this and
the other.\n */\nKotlin.Long.prototype.or = function(other) {\n return Kotlin.Long.fromBits(this.low_ |
other.low_,\n this.high_ | other.high_);\n};\n\n\n/**\n * Returns the bitwise-XOR of
this Long and the given one.\n * @param {Kotlin.Long} other The Long with which to XOR.\n * @return
{!Kotlin.Long} The bitwise-XOR of this and the other.\n */\nKotlin.Long.prototype.xor = function(other) {\n return
Kotlin.Long.fromBits(this.low_ ^ other.low_,\n this.high_ ^ other.high_);\n};\n\n\n/**\n *
Returns this Long with bits shifted to the left by the given amount.\n * @param {number} numBits The number of
bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the left by the given amount.\n
*/\nKotlin.Long.prototype.shiftLeft = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return
this;\n } else {\n var low = this.low_;\n if (numBits < 32) {\n var high = this.high_;\n return
Kotlin.Long.fromBits(\n low << numBits,\n (high << numBits) | (low >>> (32 - numBits)));\n } else
{\n return Kotlin.Long.fromBits(0, low << (numBits - 32));\n }\n }\n};\n\n\n/**\n * Returns this Long

```

```

with bits shifted to the right by the given amount.
 * @param {number} numBits The number of bits by which to
shift.
 * @return {!Kotlin.Long} This shifted to the right by the given amount.
*/
Kotlin.Long.prototype.shiftRight = function(numBits) {
 numBits &= 63;
 if (numBits == 0) {
 return this;
 } else {
 var high = this.high_;
 if (numBits < 32) {
 var low = this.low_;
 return Kotlin.Long.fromBits(
 (low >>> numBits) | (high << (32 - numBits)),
 high >> numBits);
 } else {
 return Kotlin.Long.fromBits(
 high >> (numBits - 32),
 high >= 0 ? 0 : -1);
 }
 }
};

/**
 * Returns this Long with bits shifted to the right by the given amount, with
 * zeros placed into the new leading bits.
 * @param {number} numBits The number of bits by which to shift.
 * @return {!Kotlin.Long} This shifted to the right by the given amount, with
 * zeros placed into the new leading bits.
 */
Kotlin.Long.prototype.shiftRightUnsigned
= function(numBits) {
 numBits &= 63;
 if (numBits == 0) {
 return this;
 } else {
 var high = this.high_;
 if (numBits < 32) {
 var low = this.low_;
 return Kotlin.Long.fromBits(
 (low >>> numBits) | (high << (32 - numBits)),
 high >>> numBits);
 } else if (numBits == 32) {
 return Kotlin.Long.fromBits(high, 0);
 } else {
 return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);
 }
 }
};

// Support for Kotlin
Kotlin.Long.prototype.equals = function (other) {
 return other instanceof Kotlin.Long && this.equalsLong(other);
};

Kotlin.Long.prototype.compareTo_11rb$ =
Kotlin.Long.prototype.compare;

Kotlin.Long.prototype.inc = function() {
 return this.add(Kotlin.Long.ONE);
};

Kotlin.Long.prototype.dec = function() {
 return this.add(Kotlin.Long.NEG_ONE);
};

Kotlin.Long.prototype.valueOf = function() {
 return this.toNumber();
};

Kotlin.Long.prototype.unaryPlus
= function() {
 return this;
};

Kotlin.Long.prototype.unaryMinus =
Kotlin.Long.prototype.negate;

Kotlin.Long.prototype.inv =
Kotlin.Long.prototype.not;

Kotlin.Long.prototype.rangeTo = function (other) {
 return new Kotlin.kotlin.ranges.LongRange(this, other);
};

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.
 */
Kotlin.Long.prototype.declaration = function(id, declaration) {
};

Kotlin.Long.prototype.defineInlineFunction = function(tag, fun) {
 return fun;
};

Kotlin.Long.prototype.wrapFunction = function(fun) {
 var f = function() {
 f = fun();
 return f.apply(this, arguments);
 };
 return function() {
 return f.apply(this, arguments);
 };
};

Kotlin.Long.prototype.isTypeOf = function(type) {
 return function (object) {
 return typeof object === type;
 };
};

Kotlin.Long.prototype.isInstanceOf = function (klass) {
 return function (object) {
 return Kotlin.isType(object, klass);
 };
};

Kotlin.Long.prototype.orNull = function (fn) {
 return function (object) {
 return object == null || fn(object);
 };
};

Kotlin.Long.prototype.andPredicate = function (a, b) {
 return function (object) {
 return a(object) && b(object);
 };
};

Kotlin.Long.prototype.kotlinModuleMetadata = function (abiVersion,
moduleName, data) {
};

Kotlin.Long.prototype.suspendCall = function(value) {
 return value;
};

Kotlin.Long.prototype.coroutineResult
= function(qualifier) {
 throw new Kotlin.Long.prototype.coroutineController = function(qualifier) {
 throw new Kotlin.Long.prototype.coroutineReceiver = function(qualifier) {
 throw new Kotlin.Long.prototype.setCoroutineResult = function(value, qualifier) {
 throw new Kotlin.Long.prototype.getReifiedTypeParameterKType = function(typeParameter) {
 throw new Kotlin.Long.prototype
 throw new Error(
 "This marker function should never be called. "
 + "Looks like compiler did not eliminate it properly. "
 + "Please, report an issue if you caught this
exception.");
 };

Kotlin.Long.prototype.getFunctionById = function(id, defaultValue) {
 return function() {
 return defaultValue;
 };
};

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
 */
Kotlin.Long.prototype.compareTo = function (a, b) {
 var typeA = typeof a;
 if (typeA === "number") {
 if (typeof b === "number") {
 return Kotlin.Long.prototype.doubleCompareTo(a, b);
 }
 return Kotlin.Long.prototype.primitiveCompareTo(a, b);
 }
 if (typeA === "string" || typeA === "boolean") {
 return

```



```

Kotlin.primitiveCompareTo(a, b);\n } \n return a.compareTo_11rb$(b);\n};\n\nKotlin.primitiveCompareTo
= function (a, b) {\n return a < b ? -1 : a > b ? 1 : 0;\n};\n\nKotlin.doubleCompareTo = function (a, b) {\n if (a <
b) return -1;\n if (a > b) return 1;\n\n if (a === b) {\n if (a !== 0) return 0;\n\n var ia = 1 / a;\n return
ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n } \n\n return a !== a ? (b !== b ? 0 : 1) : -1;\n};\n\nKotlin.charInc = function
(value) {\n return Kotlin.toChar(value+1);\n};\n\nKotlin.charDec = function (value) {\n return
Kotlin.toChar(value-1);\n};\n\nKotlin.imul = Math.imul || imul;\n\nKotlin.imulEmulated = imul;\n\nfunction imul(a,
b) {\n return ((a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;\n}\n\n(function() {\n var buf = new
ArrayBuffer(8);\n var bufFloat64 = new Float64Array(buf);\n var bufFloat32 = new Float32Array(buf);\n var
bufInt32 = new Int32Array(buf);\n var lowIndex = 0;\n var highIndex = 1;\n\n bufFloat64[0] = -1; //
bff00000_00000000\n
if (bufInt32[lowIndex] !== 0) {\n lowIndex = 1;\n highIndex = 0;\n } \n\n Kotlin.doubleToBits =
function(value) {\n return Kotlin.doubleToRawBits(isNaN(value) ? NaN : value);\n }; \n\n
Kotlin.doubleToRawBits = function(value) {\n bufFloat64[0] = value;\n return
Kotlin.Long.fromBits(bufInt32[lowIndex], bufInt32[highIndex]);\n }; \n\n Kotlin.doubleFromBits =
function(value) {\n bufInt32[lowIndex] = value.low_;\n bufInt32[highIndex] = value.high_;\n return
bufFloat64[0];\n }; \n\n Kotlin.floatToBits = function(value) {\n return Kotlin.floatToRawBits(isNaN(value)
? NaN : value);\n }; \n\n Kotlin.floatToRawBits = function(value) {\n bufFloat32[0] = value;\n return
bufInt32[0];\n }; \n\n Kotlin.floatFromBits = function(value) {\n bufInt32[0] = value;\n return
bufFloat32[0];\n }; \n\n // returns zero value for number with positive sign bit and non-zero value for
number with negative sign bit.\n Kotlin.doubleSignBit = function(value) {\n bufFloat64[0] = value;\n
return bufInt32[highIndex] & 0x80000000;\n }; \n\n Kotlin.numberHashCode = function(obj) {\n if ((obj | 0)
=== obj) {\n return obj | 0;\n } \n else {\n bufFloat64[0] = obj;\n return
(bufInt32[highIndex] * 31 | 0) + bufInt32[lowIndex] | 0;\n } \n } \n\n})();\n\nKotlin.ensureNotNull = function(x)
{\n return x != null ? x : Kotlin.throwNPE();\n};\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n */\n\nif (typeof String.prototype.startsWith === "undefined") {\n
Object.defineProperty(String.prototype, "startsWith", {\n value: function (searchString, position) {\n
position = position || 0;\n return this.lastIndexOf(searchString,
position) === position;\n } \n });\n\nif (typeof String.prototype.endsWith === "undefined") {\n
Object.defineProperty(String.prototype, "endsWith", {\n value: function (searchString, position) {\n var
subjectString = this.toString();\n if (position === undefined || position > subjectString.length) {\n
position = subjectString.length;\n } \n position -= searchString.length;\n var lastIndex =
subjectString.indexOf(searchString, position);\n return lastIndex !== -1 && lastIndex === position;\n } \n
});\n\n// ES6 Math polyfills\nif (typeof Math.sign === "undefined") {\n Math.sign = function(x) {\n x =
+x; // convert to a number\n if (x === 0 || isNaN(x)) {\n return Number(x);\n } \n return x > 0 ? 1
: -1;\n }; \n\nif (typeof Math.trunc === "undefined") {\n Math.trunc = function(x) {\n if (isNaN(x)) {\n
return NaN;\n } \n if (x > 0) {\n return Math.floor(x);\n } \n return Math.ceil(x);\n
}; \n\n(function() {\n var epsilon = 2.220446049250313E-16;\n var taylor_2_bound = Math.sqrt(epsilon);\n
var taylor_n_bound = Math.sqrt(taylor_2_bound);\n var upper_taylor_2_bound = 1/taylor_2_bound;\n var
upper_taylor_n_bound = 1/taylor_n_bound;\n\n if (typeof Math.sinh === "undefined") {\n Math.sinh =
function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n if (Math.abs(x) >
taylor_2_bound) {\n result += (x * x * x) / 6;\n } \n return result;\n } else {\n
var y = Math.exp(x);\n var y1 = 1 / y;\n if (!isFinite(y)) return Math.exp(x - Math.LN2);\n
if (!isFinite(y1)) return -Math.exp(-x - Math.LN2);\n return (y - y1) / 2;\n } \n
}; \n } \n\n if (typeof Math.cosh === "undefined") {\n Math.cosh = function(x) {\n var y =
Math.exp(x);\n var y1 = 1 / y;\n if (!isFinite(y) || !isFinite(y1)) return Math.exp(Math.abs(x) -
Math.LN2);\n return (y + y1) / 2;\n }; \n } \n\n if (typeof Math.tanh === "undefined") {\n
Math.tanh = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n if

```

```

(Math.abs(x) > taylor_2_bound) {\n result -= (x * x * x) / 3;\n }\n else {\n var a = Math.exp(+x), b = Math.exp(-x);\n return a === Infinity ? 1 : b ===
Infinity ? -1 : (a - b) / (a + b);\n }\n }\n }\n // Inverse hyperbolic function implementations derived
from boost special math functions,\n // Copyright Eric Ford & Hubert Holin 2001.\n if (typeof Math.asinh
=== \"undefined\") {\n var asinh = function(x) {\n if (x >= +taylor_n_bound)\n {\n if (x
> upper_taylor_n_bound)\n {\n if (x > upper_taylor_2_bound)\n {\n
// approximation by laurent series in 1/x at 0+ order from -1 to 0\n return Math.log(x) + Math.LN2;\n
 }\n else\n {\n // approximation by laurent series in 1/x at 0+ order
from -1 to 1\n return Math.log(x * 2 + (1 / (x * 2)));\n }\n }\n else\n
{\n return Math.log(x + Math.sqrt(x * x + 1));\n }\n }\n else if (x <= -
taylor_n_bound)\n {\n return -asinh(-x);\n }\n else\n {\n //
approximation by taylor
series in x at 0 up to order 2\n var result = x;\n if (Math.abs(x) >= taylor_2_bound)\n
{\n var x3 = x * x * x;\n // approximation by taylor series in x at 0 up to order 4\n
result -= x3 / 6;\n }\n return result;\n }\n }\n Math.asinh = asinh;\n }\n if
(typeof Math.acosh === \"undefined\") {\n Math.acosh = function(x) {\n if (x < 1)\n {\n
return NaN;\n }\n else if (x - 1 >= taylor_n_bound)\n {\n if (x >
upper_taylor_2_bound)\n {\n // approximation by laurent series in 1/x at 0+ order from -1 to
0\n return Math.log(x) + Math.LN2;\n }\n else\n {\n return
Math.log(x + Math.sqrt(x * x - 1));\n }\n }\n else\n {\n var y = Math.sqrt(x - 1);\n // approximation by taylor series in y at 0
up to order 2\n var result = y;\n if (y >= taylor_2_bound)\n {\n var y3 = y *
y * y;\n // approximation by taylor series in y at 0 up to order 4\n result -= y3 / 12;\n
 }\n }\n return Math.sqrt(2) * result;\n }\n }\n }\n if (typeof Math.atanh === \"undefined\")
{\n Math.atanh = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n
 if (Math.abs(x) > taylor_2_bound) {\n result += (x * x * x) / 3;\n }\n return result;\n
 }\n return Math.log((1 + x) / (1 - x)) / 2;\n }\n }\n if (typeof Math.log1p === \"undefined\") {\n
Math.log1p = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var x2 = x * x;\n var x3 = x2 * x;\n var x4
= x3 * x;\n // approximation by taylor series in x at 0 up to order 4\n return (-x4 / 4 + x3 / 3 - x2 /
2 + x);\n }\n return Math.log(x + 1);\n }\n }\n if (typeof Math.expm1 === \"undefined\") {\n
Math.expm1 = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var x2 = x * x;\n
 var x3 = x2 * x;\n var x4 = x3 * x;\n // approximation by taylor series in x at 0 up to order 4\n
 return (x4 / 24 + x3 / 6 + x2 / 2 + x);\n }\n return Math.exp(x) - 1;\n }\n }\n }\n if (typeof
Math.hypot === \"undefined\") {\n Math.hypot = function() {\n var y = 0;\n var length =
arguments.length;\n for (var i = 0; i < length; i++) {\n if (arguments[i]
=== Infinity || arguments[i] === -Infinity) {\n return Infinity;\n }\n y += arguments[i] *
arguments[i];\n }\n return Math.sqrt(y);\n }\n }\n if (typeof Math.log10 === \"undefined\") {\n
Math.log10 = function(x) {\n return Math.log(x) * Math.LOG10E;\n }\n }\n if (typeof Math.log2 ===
\"undefined\") {\n Math.log2 = function(x) {\n return Math.log(x) * Math.LOG2E;\n }\n }\n if (typeof
Math.clz32 === \"undefined\") {\n Math.clz32 = (function(log, LN2) {\n return function(x) {\n var
asUint = x >>> 0;\n if (asUint === 0) {\n return 32;\n }\n return 31 - (log(asUint) /
LN2 | 0) | 0; // the \"| 0\" acts like math.floor\n }\n })(Math.log, Math.LN2);\n }\n }\n // For HtmlUnit and
PhantomJs\n if (typeof ArrayBuffer.isView === \"undefined\") {\n ArrayBuffer.isView = function(a) {\n
return a != null && a.__proto__ != null && a.__proto__.__proto__
=== Int8Array.prototype.__proto__;\n }\n }\n }\n if (typeof Array.prototype.fill === \"undefined\") {\n // Polyfill
from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n
Object.defineProperty(Array.prototype, 'fill', {\n value: function (value) {\n // Steps 1-2.\n if

```



```

null && \"value\" in propertyDescriptor) {\n thisObject[propertyName] = value;\n return\n }\n\nKotlin.callSetter(thisObject, Object.getPrototypeOf(klass), propertyName, value);\n};\n\nfunction
isInheritanceFromInterface(ctor, iface) {\n if (ctor === iface) return true;\n var metadata = ctor.$metadata$;\n if (metadata != null) {\n var interfaces = metadata.interfaces;\n for (var i = 0; i < interfaces.length; i++)
{\n if (isInheritanceFromInterface(interfaces[i], iface)) {\n return true;\n }\n }\n }\n var superPrototype = ctor.prototype != null ? Object.getPrototypeOf(ctor.prototype) : null;\n var superConstructor
= superPrototype != null ? superPrototype.constructor : null;\n return superConstructor != null &&
isInheritanceFromInterface(superConstructor, iface);\n}\n\n/**\n * @param {*} object\n * @param
{Function|Object} klass\n * @returns {Boolean}\n */\nKotlin.isType = function (object, klass) {\n if (klass ===
Object) {\n switch (typeof object) {\n case \"string\":\n case \"number\":\n case
\"boolean\":\n case \"function\":\n return true;\n default:\n return object instanceof
Object;\n }\n }\n\n if (object == null || klass == null || (typeof object !== 'object' && typeof object !== 'function')) {\n return
false;\n }\n\n if (typeof klass === \"function\" && object instanceof klass) {\n return true;\n }\n\n var
proto = Object.getPrototypeOf(klass);\n var constructor = proto != null ? proto.constructor : null;\n if
(constructor != null && \"\$metadata$\" in constructor) {\n var metadata = constructor.$metadata$;\n if
(metadata.kind === Kotlin.Kind.OBJECT) {\n return object === klass;\n }\n }\n\n var classMetadata
= klass.$metadata$;\n\n // In WebKit (JavaScriptCore) for some interfaces from DOM typeof returns \"object\",
nevertheless they can be used in RHS of instanceof\n if (classMetadata == null) {\n return object instanceof
klass;\n }\n\n if (classMetadata.kind === Kotlin.Kind.INTERFACE && object.constructor != null) {\n
return isInheritanceFromInterface(object.constructor, klass);\n }\n\n return false;\n};\n\nKotlin.isNumber = function (a) {\n return typeof a == \"number\" || a instanceof
Kotlin.Long;\n};\n\nKotlin.isChar = function (value) {\n return value instanceof
Kotlin.BoxedChar;\n};\n\nKotlin.isComparable = function (value) {\n var type = typeof value;\n\n return type
=== \"string\" ||\n type === \"boolean\" ||\n Kotlin.isNumber(value) ||\n Kotlin.isType(value,
Kotlin.kotlin.Comparable);\n};\n\nKotlin.isCharSequence = function (value) {\n return typeof value === \"string\"
|| Kotlin.isType(value, Kotlin.kotlin.CharSequence);\n};\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the
module\n\n@PublishedApi\nexternal fun <T> Array(size: Int): Array<T>\n\n@JsName(\"newArray\")\nfun
<T> newArray(size: Int, initialValue: T) = fillArrayVal(Array<T>(size),
initialValue)\n\n@JsName(\"newArrayF\")\ninline fun <T> arrayWithFun(size: Int, init: (Int) -> T) =
fillArrayFun(Array<T>(size), init)\n\n@JsName(\"fillArray\")\ninline fun <T> fillArrayFun(array: Array<T>, init:
(Int) -> T): Array<T> {\n for (i in 0..array.size - 1) {\n array[i] = init(i)\n }\n return
array\n}\n\n@JsName(\"booleanArray\")\nfun booleanArray(size: Int, init: dynamic): Array<Boolean> {\n val
result: dynamic = Array<Boolean>(size)\n result.`$type$` = \"BooleanArray\"\n return when (init) {\n null,
true -> fillArrayVal(result, false)\n false -> result\n else -> fillArrayFun<Boolean>(result, init)\n }\n}\n\n@JsName(\"booleanArrayF\")\ninline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean):
Array<Boolean> = fillArrayFun(booleanArray(size, false),
init)\n\n@JsName(\"charArray\")\n@Suppress(\"UNUSED_PARAMETER\")\nfun charArray(size: Int, init:
dynamic):
Array<Char> {\n val result = js(\"new Uint16Array(size)\")\n result.`$type$` = \"CharArray\"\n return when
(init) {\n null, true, false -> result // For consistency\n else -> fillArrayFun<Char>(result, init)\n }\n}\n\n@JsName(\"charArrayF\")\ninline fun charArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {\n
val array = charArray(size, null)\n for (i in 0..array.size - 1) {\n @Suppress(\"UNUSED_VARIABLE\") //
used in js block\n val value = init(i)\n js(\"array[i] = value;\")\n }\n return
array\n}\n\n@JsName(\"untypedCharArrayF\")\ninline fun untypedCharArrayWithFun(size: Int, init: (Int) -> Char):
Array<Char> {\n val array = Array<Char>(size)\n for (i in 0..array.size - 1) {\n

```



```

@JsName("SIZE_BITS")\n const val SIZE_BITS = 8\n}\n\n@JsName("CharCompanionObject")\ninternal
object CharCompanionObject {\n @JsName("MIN_VALUE")\n public const val MIN_VALUE: Char =
"\u0000"\n\n @JsName("MAX_VALUE")\n public const val MAX_VALUE: Char = "\uFFFF"\n\n
@JsName("MIN_HIGH_SURROGATE")\n public const val MIN_HIGH_SURROGATE: Char = "\uD800"\n\n
@JsName("MAX_HIGH_SURROGATE")\n public const val MAX_HIGH_SURROGATE: Char =
"\uDBFF"\n\n @JsName("MIN_LOW_SURROGATE")\n public const val MIN_LOW_SURROGATE: Char =
"\uDC00"\n\n @JsName("MAX_LOW_SURROGATE")\n public const val MAX_LOW_SURROGATE: Char
= "\uDFFF"\n\n
 @JsName("MIN_SURROGATE")\n public const val MIN_SURROGATE: Char =
MIN_HIGH_SURROGATE\n\n @JsName("MAX_SURROGATE")\n public const val MAX_SURROGATE:
Char = MAX_LOW_SURROGATE\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 2\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 16\n}\n\ninternal object StringCompanionObject
{\n}\n\ninternal object BooleanCompanionObject {\n}\n\n",/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element*
from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in
Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
Array<out T>.component1(): T {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If
the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component1():
Byte {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less
than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component1(): Short {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less
than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component1(): Int {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component1(): Long {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component1(): Float {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n
* where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun
DoubleArray.component1(): Double {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n *
If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the
behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component1():
Boolean {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is
less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component1(): Char {\n return
get(0)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic

```



```

*^@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component3(): Boolean {\n return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component3(): Char {\n return
get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component4(): T {\n return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline
operator fun ByteArray.component4(): Byte {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the
array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsExcep] except in Kotlin/JS\n
* where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun
ShortArray.component4(): Short {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If
the size of this array is less than 4, throws an [IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the
behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component4(): Int
{\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4,
throws an [IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component4(): Long {\n return
get(3)\n}\n\n/**\n
* Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component4(): Float {\n return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component4(): Double {\n return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component4(): Boolean {\n return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n
* \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsExcep] except in Kotlin/JS\n * where
the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component4():
Char {\n return get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less
than 5, throws an [IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component5(): T {\n return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component5(): Byte {\n return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component5(): Short {\n return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component5(): Int {\n return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*^@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component5(): Long {\n return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an

```



[IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

*\/@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component5(): Float {
 return get(4)\n}\n\n/**
 * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5,
 throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component5(): Double {
 return
get(4)\n}\n\n/**
 * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component5(): Boolean {
 return
get(4)\n}\n\n/**
 * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic
inline operator fun CharArray.component5(): Char {
 return get(4)\n}\n\n/**
 * Returns `true` if [element] is
found in the array.\n */\npublic operator fun <\/@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element:
T): Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true` if [element] is found in the array.\n
*/\npublic operator fun ByteArray.contains(element: Byte): Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true` if [element] is found in the array.\n */\npublic operator fun ShortArray.contains(element: Short):
Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true` if [element] is found in the array.\n
*/\npublic operator fun IntArray.contains(element: Int): Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true` if
[element] is found in the array.\n */\npublic operator fun LongArray.contains(element: Long):
Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true` if
[element] is found in the array.\n */\n@Deprecated("The function has unclear behavior when searching for NaN or
zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or
.asList().contains(element: T)' to get the same search behavior as in a list.", ReplaceWith("any { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince = "1.7")\npublic
operator fun FloatArray.contains(element: Float): Boolean {
 return any { it == element }\n}\n\n/**
 * Returns
`true` if [element] is found in the array.\n */\n@Deprecated("The function has unclear behavior when searching for
NaN or zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or
.asList().contains(element: T)' to get the same search behavior as in a list.", ReplaceWith("any { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince
= "1.7")\npublic operator fun DoubleArray.contains(element: Double): Boolean {
 return any { it == element
}\n}\n\n/**
 * Returns `true` if [element] is found in the array.\n */\npublic operator fun
BooleanArray.contains(element: Boolean): Boolean {
 return indexOf(element) >= 0\n}\n\n/**
 * Returns `true`
if [element] is found in the array.\n */\npublic operator fun CharArray.contains(element: Char): Boolean {
 return
indexOf(element) >= 0\n}\n\n/**
 * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\npublic expect fun <\/@kotlin.internal.OnlyInputTypes T> Array<out T>.elementAt(index:
Int): T\n\n/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
expect fun ByteArray.elementAt(index: Int):
Byte\n\n/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
expect fun ShortArray.elementAt(index: Int): Short\n\n/**
 * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\npublic expect fun IntArray.elementAt(index: Int):
Int\n\n/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
expect fun LongArray.elementAt(index: Int): Long\n\n/**
 * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample

```

```

samples.collections.Collections.Elements.elementAt\n
*\npublic expect fun FloatArray.elementAt(index: Int): Float\n\n**\n * Returns an element at the given [index] or
throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\npublic expect fun DoubleArray.elementAt(index: Int):
Double\n\n**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\npublic expect fun BooleanArray.elementAt(index: Int): Boolean\n\n**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\npublic expect fun CharArray.elementAt(index: Int):
Char\n\n**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function
if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) ->
Short): Short {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n**\n *
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int):
Int {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n**\n * Returns an
element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this
array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.elementAtOrElse(index: Int, defaultValue:
(Int) -> Long): Long {\n return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Float): Float {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Double): Double {\n return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n**\n
* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) ->
Boolean): Boolean {\n return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n**\n * Returns an element at the given [index] or `null` if the

```

[index] is out of bounds of this

```
array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.elementAtOrNull(index: Int): T? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.elementAtOrNull(index: Int): Byte? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.elementAtOrNull(index: Int): Short? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.elementAtOrNull(index: Int): Int? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.elementAtOrNull(index: Int): Long? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.elementAtOrNull(index: Int): Float? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.elementAtOrNull(index: Int): Double? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds\n of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.elementAtOrNull(index: Int): Boolean? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of\n bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.elementAtOrNull(index: Int): Char? {\n return\n this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such\n element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic\n\n inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n ByteArray.find(predicate: (Byte) -> Boolean): Byte? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the\n first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n ShortArray.find(predicate: (Short) -> Boolean): Short? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the\n first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n IntArray.find(predicate: (Int) ->\n\n Boolean): Int? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the first element matching the given\n [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n LongArray.find(predicate: (Long) -> Boolean): Long? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the\n first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n FloatArray.find(predicate: (Float) -> Boolean): Float? {\n return\n firstOrNull(predicate)\n }\n\n/**\n * Returns the\n first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
```

```

samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.find(predicate: (Double) -> Boolean): Double? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.find(predicate: (Char) -> Boolean): Char? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.findLast(predicate: (Short) -> Boolean): Short? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.findLast(predicate: (Int) -> Boolean): Int? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last
element matching the given [predicate],
or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {\n
return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {\n
return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double?
{\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if
no such element was
found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {\n return
lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {\n
return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if
the array is empty.\n */\npublic fun <T> Array<out T>.first(): T {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n */\npublic fun ByteArray.first(): Byte {\n if
(isEmpty())\n throw NoSuchElementException("Array
is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException
if the array is empty.\n */\npublic fun ShortArray.first(): Short {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n */\npublic fun IntArray.first(): Int {\n if
(isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the
first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun LongArray.first(): Long {\n if
(isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the

```

```

first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun FloatArray.first():\nFloat {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is\n * empty.\n */\npublic fun DoubleArray.first(): Double {\n if (isEmpty())\n throw\n NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun BooleanArray.first(): Boolean {\n if\n (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the\n * first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun CharArray.first():\nChar {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such\n * element is found.\n */\npublic inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {\n for\n (element in this) if (predicate(element)) return element\n throw NoSuchElementException("Array contains no\n element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun ByteArray.first(predicate:\n(Byte) -> Boolean): Byte {\n for (element in this) if (predicate(element)) return element\n throw\n NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first\n * element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {\n for (element in this) if\n (predicate(element)) return element\n throw NoSuchElementException("Array contains no element matching the\n predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws\n * [NoSuchElementException]\n * if no such element is found.\n */\npublic inline fun IntArray.first(predicate: (Int) -> Boolean): Int {\n for (element\n in this) if (predicate(element)) return element\n throw NoSuchElementException("Array contains no element\n matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws\n * [NoSuchElementException] if no such element is found.\n */\npublic inline fun LongArray.first(predicate: (Long) -\n> Boolean): Long {\n for (element in this) if (predicate(element)) return element\n throw\n NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first\n * element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {\n for (element in this) if\n (predicate(element)) return element\n throw NoSuchElementException("Array contains no element matching the\n predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such\n * element is found.\n */\npublic inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {\n for\n (element in this) if (predicate(element)) return element\n throw NoSuchElementException("Array contains no\n element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws\n * [NoSuchElementException] if no such element is found.\n */\npublic inline fun\nBooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (predicate(element))\n return element\n throw NoSuchElementException("Array contains no element matching the\n predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws\n * [NoSuchElementException] if no such element is found.\n */\npublic inline fun CharArray.first(predicate: (Char) ->\nBoolean): Char {\n for (element in this)\n if (predicate(element)) return element\n throw NoSuchElementException("Array contains no element matching\n the predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to\n * elements of this array in iteration order,\n * \n * or throws [NoSuchElementException] if no non-null value was\n * produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out\nT>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw\n NoSuchElementException("No element of the array was transformed to a non-null value.")\n}\n\n/**\n * Returns

```



```

ByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n return if (index >=
 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index]
or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.getOrElse(index: Int, defaultValue: (Int) -> Short):
Short {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of
this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.getOrElse(index: Int, defaultValue: (Int) ->
Int): Int {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n *
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.getOrElse(index:
 Int, defaultValue: (Int) -> Long): Long {\n return if (index >= 0 && index <= lastIndex) get(index) else
 defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.getOrElse(index: Int, defaultValue: (Int) -> Float): Float {\n return if (index >= 0 && index <=
 lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.getOrElse(index: Int, defaultValue: (Int) ->
Double): Double {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index]
is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.getOrElse(index:
 Int, defaultValue: (Int) -> Boolean): Boolean {\n return if (index >= 0 && index <= lastIndex) get(index) else
 defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n return if (index >= 0 && index <=
 lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n
*/\npublic fun <T> Array<out T>.getOrNull(index: Int): T? {\n return if (index >= 0 && index <= lastIndex)
 get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic
fun ByteArray.getOrNull(index: Int): Byte? {\n return if (index >= 0 && index <= lastIndex) get(index) else
 null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n
* @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun ShortArray.getOrNull(index: Int):
Short? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n */\npublic fun IntArray.getOrNull(index: Int): Int? {\n
 return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*/\npublic fun LongArray.getOrNull(index: Int): Long? {\n return if (index >= 0 && index <= lastIndex)
 get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of
this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun
FloatArray.getOrNull(index: Int): Float? {\n return if (index >= 0 && index <= lastIndex) get(index) else
 null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n
* @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun DoubleArray.getOrNull(index:
 Int): Double? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n */\npublic fun BooleanArray.getOrNull(index:

```

```

Int): Boolean? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n */\npublic fun CharArray.getOrNull(index: Int): Char? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of [element], or -
1 if the array does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Array<out
T>.indexOf(element: T): Int {\n if (element == null) {\n for (index in indices) {\n if (this[index] ==
null) {\n return index\n }\n }\n } else {\n for (index in indices) {\n if (element ==
this[index]) {\n return index\n }\n }\n }\n return -1\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n */\npublic fun ByteArray.indexOf(element: Byte): Int {\n for (index in indices) {\n if (element ==
this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if
the array does not contain element.\n */\npublic fun ShortArray.indexOf(element: Short): Int {\n for (index in
indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n */\npublic fun IntArray.indexOf(element: Int):
Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -
1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun
LongArray.indexOf(element: Long): Int {\n for (index in indices) {\n if (element == this[index]) {\n
return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n
*/\n\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed
soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)
to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince = "1.7")\npublic fun
FloatArray.indexOf(element: Float): Int {\n for (index in indices) {\n if (element == this[index]) {\n
return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not
contain element.\n */\n\n@Deprecated("The function has unclear behavior when searching for NaN or zero values
and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this
behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.",
ReplaceWith("indexOfFirst { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6", hiddenSince = "1.7")\npublic fun DoubleArray.indexOf(element: Double): Int {\n for (index in indices)
{\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first
index of [element], or -1 if the array does not contain element.\n */\n\npublic fun BooleanArray.indexOf(element:
Boolean): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n
return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n\npublic fun
CharArray.indexOf(element: Char): Int {\n for (index in indices) {\n if (element == this[index]) {\n
return index\n }\n }\n return -1\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*/\n\npublic inline fun <T> Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n for (index in indices) {\n
if (predicate(this[index])) {\n return index\n }\n return -1\n}\n\n/**\n * Returns index of the
first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun
ByteArray.indexOfFirst(predicate: (Byte) -> Boolean): Int {\n for (index in indices) {\n if
(predicate(this[index])) {\n return index\n }\n return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\n\npublic inline fun
ShortArray.indexOfFirst(predicate: (Short) -> Boolean): Int {\n for (index in indices) {\n if
(predicate(this[index])) {\n return index\n }\n return -1\n}\n\n/**\n * Returns index of the first element
matching the given [predicate], or -1 if the array
does not contain such element.\n */\n\npublic inline fun IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int {\n
for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -

```



```

I\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain
such element.\n */\npublic inline fun LongArray.indexOfFirst(predicate: (Long) -> Boolean): Int {\n for (index in
indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*/\npublic inline fun FloatArray.indexOfFirst(predicate: (Float) -> Boolean): Int {\n for (index in indices) {\n
if (predicate(this[index])) {\n
 return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given
[predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
DoubleArray.indexOfFirst(predicate: (Double) -> Boolean): Int {\n for (index in indices) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
BooleanArray.indexOfFirst(predicate: (Boolean) -> Boolean): Int {\n for (index in indices) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int {\n for (index in indices) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T>
Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int {\n for (index in indices.reversed()) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int {\n for (index in indices.reversed()) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ShortArray.indexOfLast(predicate: (Short) -> Boolean):
Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n
return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not
contain such element.\n */\npublic inline fun IntArray.indexOfLast(predicate: (Int) -> Boolean): Int {\n for (index
in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -
1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain
such element.\n */\npublic inline fun LongArray.indexOfLast(predicate: (Long) -> Boolean): Int {\n for (index in
indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such
element.\n */\npublic inline
fun FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {\n for (index in indices.reversed()) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
DoubleArray.indexOfLast(predicate: (Double) -> Boolean): Int {\n for (index in indices.reversed()) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
BooleanArray.indexOfLast(predicate: (Boolean) -> Boolean): Int {\n for (index in indices.reversed()) {\n if
(predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or
-1 if the array does not contain such element.\n */\npublic inline fun CharArray.indexOfLast(predicate: (Char) ->
Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n
 }\n }\n return -1\n}\n\n/**\n * Returns the last element.\n * @throws NoSuchElementException if the array
is empty.\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.last():
T {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * @throws NoSuchElementException if the array is

```

```

empty.\n * \n * @sample samples.collections.Collections.Elements.last\n *^\npublic fun ByteArray.last(): Byte {\n
if (isEmpty())\n throw NoSuchElementException("Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n *
Returns the last element.\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n * @sample samples.collections.Collections.Elements.last\n *^\npublic fun ShortArray.last(): Short {\n
if (isEmpty())\n throw NoSuchElementException("Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n *
Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic fun IntArray.last(): Int {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n *
\n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic fun LongArray.last(): Long {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*^\npublic fun FloatArray.last(): Float {\n if (isEmpty())\n throw NoSuchElementException("Array is
empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*^\npublic fun DoubleArray.last(): Double {\n if (isEmpty())\n throw NoSuchElementException("Array is
empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*^\npublic fun BooleanArray.last(): Boolean {\n if (isEmpty())\n throw NoSuchElementException("Array is
empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*^\npublic fun CharArray.last():
Char {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.\")\n return
this[lastIndex]\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun <T> Array<out T>.last(predicate: (T) ->
Boolean): T {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n throw NoSuchElementException("Array contains no element
matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ByteArray.last(predicate: (Byte) -> Boolean):
Byte {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n throw NoSuchElementException("Array contains no element
matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ShortArray.last(predicate: (Short) -> Boolean):
Short {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element))
return element\n }\n throw NoSuchElementException("Array contains no element matching the
predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun IntArray.last(predicate: (Int) -> Boolean): Int
{\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element))
return element\n }\n throw NoSuchElementException("Array contains no element
matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun LongArray.last(predicate: (Long) -> Boolean):
Long {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element))
return element\n }\n throw NoSuchElementException("Array contains no element
matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample

```

```

return element\n } throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.last(predicate: (Float) -> Boolean):
Float {\n for (index in this.indices.reversed()) {\n val element =
this[index]\n if (predicate(element)) return element\n } throw NoSuchElementException("Array contains
no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun DoubleArray.last(predicate: (Double) ->
Boolean): Double {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n } throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun BooleanArray.last(predicate: (Boolean) ->
Boolean): Boolean {\n for (index in this.indices.reversed())
{\n val element = this[index]\n if (predicate(element)) return element\n } throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
* \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun CharArray.last(predicate:
(Char) -> Boolean): Char {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n } throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain
element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.lastIndexOf(element: T): Int {\n if
(element == null) {\n for (index in indices.reversed()) {\n if (this[index] == null) {\n
return index\n } }\n } else {\n for (index in indices.reversed()) {\n if (element ==
this[index]) {\n return index\n } }\n }\n return -1\n}\n\n/**\n * Returns last index of
[element], or -1 if the array does not contain element.\n */\npublic fun ByteArray.lastIndexOf(element: Byte): Int
{\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n } }\n
}\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n
*/\npublic fun ShortArray.lastIndexOf(element: Short): Int {\n for (index in indices.reversed()) {\n if (element ==
this[index]) {\n return index\n } }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if
the array does not contain element.\n */\npublic fun IntArray.lastIndexOf(element: Int): Int {\n for (index in
indices.reversed())
{\n if (element == this[index]) {\n return index\n } }\n }\n return -1\n}\n\n/**\n * Returns last
index of [element], or -1 if the array does not contain element.\n */\npublic fun LongArray.lastIndexOf(element:
Long): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n
}\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n
*/\n@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed
soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element:
T)' to get the same search behavior as in a list.", ReplaceWith("indexOfLast { it == element
}"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince = "1.7")\npublic fun
FloatArray.lastIndexOf(element: Float): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n } }\n }\n return -1\n}\n\n/**\n * Returns last
index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("The function has unclear
behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }'
instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a
list.", ReplaceWith("indexOfLast { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.6", hiddenSince = "1.7")\npublic fun DoubleArray.lastIndexOf(element: Double): Int {\n for

```

```

(index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -
1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
BooleanArray.lastIndexOf(element: Boolean): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last
index of [element], or -1 if the array does not contain element.\n */\npublic fun CharArray.lastIndexOf(element:
Char): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.lastOrNull(): T? {\n return if
(isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.lastOrNull(): Byte? {\n return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.lastOrNull(): Short? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun IntArray.lastOrNull(): Int? {\n return if (isEmpty())
null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun LongArray.lastOrNull(): Long? {\n return if
(isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun FloatArray.lastOrNull(): Float? {\n return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.lastOrNull():
Double? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the
array is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun
BooleanArray.lastOrNull(): Boolean? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last
element, or `null` if the array is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic
fun CharArray.lastOrNull(): Char? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Array<out T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun ByteArray.lastOrNull(predicate: (Byte) ->
Boolean): Byte? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun ShortArray.lastOrNull(predicate: (Short) ->
Boolean): Short? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun IntArray.lastOrNull(predicate:
(Int) -> Boolean): Int? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun LongArray.lastOrNull(predicate: (Long) ->
Boolean): Long? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.lastOrNull(predicate: (Float) ->

```



```

random element from this array using the specified source of randomness.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun IntArray.random(random:
Random): Int {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns
a random element from this array using the specified source of randomness.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun
LongArray.random(random: Random): Long {\n if (isEmpty())\n throw NoSuchElementException("Array is
empty.")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the
specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin("1.3")\npublic fun FloatArray.random(random: Random): Float {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun DoubleArray.random(random: Random): Double {\n
 if (isEmpty())\n throw NoSuchElementException("Array
is empty.")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the
specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin("1.3")\npublic fun BooleanArray.random(random: Random): Boolean {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a
random element from this array using the specified source of randomness.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Array<out T>.randomOrNull(): T? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.randomOrNull(): Byte? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.randomOrNull(): Short? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.randomOrNull(): Int? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is
empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.randomOrNull(): Long? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.randomOrNull(): Float? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.randomOrNull(): Double? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.randomOrNull():

```

```

Boolean? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if
this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.randomOrNull(): Char? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array using the specified source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out
T>.randomOrNull(random: Random): T? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ByteArray.randomOrNull(random: Random): Byte? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ShortArray.randomOrNull(random: Random): Short? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntArray.randomOrNull(random: Random): Int? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongArray.randomOrNull(random: Random): Long? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using
the specified source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
FloatArray.randomOrNull(random: Random): Float? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
DoubleArray.randomOrNull(random: Random): Double? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
BooleanArray.randomOrNull(random:
Random): Boolean? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n *
Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharArray.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n
*\npublic fun <T> Array<out T>.single(): T {\n return when (size) {\n 0 ->
throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\npublic fun ByteArray.single(): Byte
{\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else
-> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single
element, or throws an exception if the array is empty or has more than one element.\n
*\npublic fun

```

```

ShortArray.single(): Short {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun IntArray.single(): Int {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun LongArray.single(): Long {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun FloatArray.single(): Float {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun DoubleArray.single(): Double {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun BooleanArray.single(): Boolean {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element, or throws an exception if the array is empty or has more than one element. */
public fun CharArray.single(): Char {
 return when (size) {
 0 -> throw NoSuchElementException("Array is empty.")
 1 -> this[0]
 else -> throw IllegalArgumentException("Array has more than one element.")
 }
}

/** Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element. */
public inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {
 var single: T? = null
 var found = false
 for (element in this) {
 if (predicate(element)) {
 if (found) throw IllegalArgumentException("Array contains more than one matching element.")
 single = element
 found = true
 }
 }
 if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
 @SuppressWarnings("UNCHECKED_CAST") return single as T
}

/** Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element. */
public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
 var single: Byte? = null
 var found = false
 for (element in this) {
 if (predicate(element)) {
 if (found) throw IllegalArgumentException("Array contains more than one matching element.")
 single = element
 found = true
 }
 }
 if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
 @SuppressWarnings("UNCHECKED_CAST") return single as Byte
}

/** Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element. */
public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
 var single: Short? = null
 var found = false
 for (element in this) {
 if (predicate(element)) {
 if (found) throw IllegalArgumentException("Array contains more than one matching element.")
 single = element
 found = true
 }
 }
 if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
 @SuppressWarnings("UNCHECKED_CAST") return single as Short
}

/** Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element. */
public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
 var single: Int? = null
 var found = false
 for (element in this) {
 if (predicate(element)) {
 if (found) throw IllegalArgumentException("Array contains more than one matching element.")
 single = element
 found = true
 }
 }
 if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
 @SuppressWarnings("UNCHECKED_CAST") return single as Int
}

```



```

the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun
LongArray.single(predicate: (Long) -> Boolean): Long {\n var single: Long? = null\n var found = false\n for
(element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no element matching the
predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as Long\n}\n\n/**\n * Returns the single
element matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*/\npublic inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {\n var single: Float? = null\n var
found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw
IllegalArgumentException("Array contains more than one matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as Float\n}\n\n/**\n * Returns the
single element matching the given [predicate], or throws exception if there is no or more than one matching
element.\n */\npublic inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {\n var single:
Double? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found)
throw IllegalArgumentException("Array contains more than one matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no element
matching the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as Double\n}\n\n/**\n * Returns the
single element matching the given [predicate], or throws exception if there is no or more than one
matching element.\n */\npublic inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {\n var
single: Boolean? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if
(found) throw IllegalArgumentException("Array contains more than one
matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw
NoSuchElementException("Array contains no element matching the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as Boolean\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic
inline fun CharArray.single(predicate: (Char) -> Boolean): Char {\n var single: Char? = null\n var found =
false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw
IllegalArgumentException("Array contains more than one matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as Char\n}\n\n/**\n * Returns single
element, or `null` if the
array is empty or has more than one element.\n */\npublic fun <T> Array<out T>.singleOrNull(): T? {\n return if
(size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one
element.\n */\npublic fun ByteArray.singleOrNull(): Byte? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun
ShortArray.singleOrNull(): Short? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element,
or `null` if the array is empty or has more than one element.\n */\npublic fun IntArray.singleOrNull(): Int? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n */\npublic fun LongArray.singleOrNull(): Long? {\n return if (size == 1) this[0] else
null\n}\n\n/**\n * Returns single element, or `null` if the array is empty
or has more than one element.\n */\npublic fun FloatArray.singleOrNull(): Float? {\n return if (size == 1) this[0]
else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n
*/\npublic fun DoubleArray.singleOrNull(): Double? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun
BooleanArray.singleOrNull(): Boolean? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n */\npublic fun CharArray.singleOrNull():
Char? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element matching the given

```

```

[predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun <T>
Array<out T>.singleOrNull(predicate: (T) -> Boolean): T? {\n var single: T? = null\n var found
= false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n */\npublic inline fun ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? {\n var single: Byte?
= null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return
null\n single = element\n found = true\n }\n }\n if (!found) return null\n return
single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or
more than one element was found.\n */\npublic inline fun ShortArray.singleOrNull(predicate: (Short) -> Boolean):
Short? {\n var single: Short? = null\n
 var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n
single = element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n
* Returns the single element matching the given [predicate], or `null` if element was not found or more than one
element was found.\n */\npublic inline fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? {\n var single:
Int? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found)
return
null\n single = element\n found = true\n }\n }\n if (!found) return null\n return
single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or
more than one element was found.\n */\npublic inline fun LongArray.singleOrNull(predicate: (Long) -> Boolean):
Long? {\n var single: Long? =
null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return
null\n single = element\n found = true\n }\n }\n if (!found) return null\n return
single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or
more than one element was found.\n */\npublic inline fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float?
{\n var
single: Float? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if
(found) return
null\n single = element\n found = true\n }\n }\n if (!found) return null\n return
single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or
more than one element was found.\n */\npublic inline fun DoubleArray.singleOrNull(predicate: (Double) ->
Boolean): Double? {\n
 var single: Double? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n
 if (found) return
null\n single = element\n found = true\n }\n }\n if (!found) return null\n return
single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or
more than one element was found.\n */\npublic inline fun BooleanArray.singleOrNull(predicate: (Boolean) ->
Boolean): Boolean? {\n var single: Boolean? = null\n var found = false\n for (element in this) {\n if
(predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n
 }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or
`null` if element was not found or more than one element was found.\n */\npublic inline fun
CharArray.singleOrNull(predicate: (Char) ->
Boolean): Char? {\n var single: Char? = null\n var found = false\n for (element in this) {\n if
(predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n
 }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list containing all elements except first [n]
elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Array<out T>.drop(n: Int): List<T> {\n
 require(n >= 0) {\n "Requested element count $n is less than zero.\n }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.drop(n: Int): List<Byte> {\n
 require(n >= 0) {\n "Requested element count

```

```

 $n is less than zero." } }n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all
elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun ShortArray.drop(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun IntArray.drop(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n
* \n * @sample samples.collections.Collections.Transformations.drop\n * \npublic fun LongArray.drop(n: Int):
List<Long> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun FloatArray.drop(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun DoubleArray.drop(n: Int): List<Double> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return
takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n
* \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun BooleanArray.drop(n: Int): List<Boolean>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun CharArray.drop(n: Int): List<Char> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
* \npublic fun <T> Array<out T>.dropLast(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is
less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements
except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun ByteArray.dropLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun ShortArray.dropLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n]
is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n * \npublic fun
IntArray.dropLast(n: Int): List<Int> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n
return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n]
elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic fun LongArray.dropLast(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -

```

```

n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic
fun FloatArray.dropLast(n: Int): List<Float> {\n require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun DoubleArray.dropLast(n: Int): List<Double>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun BooleanArray.dropLast(n: Int):
List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing
all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.drop\n *\npublic fun CharArray.dropLast(n: Int):
List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic inline fun <T>
Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if
(!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns
a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun ByteArray.dropLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun ShortArray.dropLastWhile(predicate:
(Short) -> Boolean): List<Short> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun IntArray.dropLastWhile(predicate:
(Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all
elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun LongArray.dropLastWhile(predicate:
(Long) -> Boolean): List<Long> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun FloatArray.dropLastWhile(predicate:
(Float) -> Boolean): List<Float> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except
last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun DoubleArray.dropLastWhile(predicate:
(Double) -> Boolean): List<Double> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
 return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun
BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n for (index in lastIndex

```

```

downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n } return
emptyList()\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {\n for (index in
lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n } return
emptyList()\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun <T>
Array<out T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n var yielding = false\n val list =
ArrayList<T>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements
except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun ByteArray.dropWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n var yielding = false\n val list = ArrayList<Byte>()\n for (item in this)\n
if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding =
true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun
ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {\n var yielding = false\n val list =
ArrayList<Short>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item))
{\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n var yielding = false\n val
list = ArrayList<Int>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item))
{\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun LongArray.dropWhile(predicate:
(Long) -> Boolean): List<Long> {\n var yielding = false\n val list = ArrayList<Long>()\n for (item in this)\n
if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding =
true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n *
\n * @sample samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun
FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n var yielding = false\n val list =
ArrayList<Float>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item))
{\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun DoubleArray.dropWhile(predicate:
(Double) -> Boolean): List<Double> {\n var yielding = false\n val list = ArrayList<Double>()\n for (item in
this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n
yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements
except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*/\npublic inline fun BooleanArray.dropWhile(predicate:
(Boolean) -> Boolean): List<Boolean> {\n var yielding = false\n val list = ArrayList<Boolean>()\n for (item in
this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n
yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*/\npublic
inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {\n var yielding = false\n val list =
ArrayList<Char>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n
 list.add(item)\n yielding = true\n }\n }\n}

```

```

return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun <T> Array<out T>.filter(predicate: (T) ->
Boolean): List<T> {\n return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only
elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\npublic inline fun ByteArray.filter(predicate: (Byte) -> Boolean): List<Byte> {\n return
filterTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun
ShortArray.filter(predicate: (Short) -> Boolean): List<Short> {\n return filterTo(ArrayList<Short>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\npublic inline fun IntArray.filter(predicate: (Int) -> Boolean): List<Int> {\n return filterTo(ArrayList<Int>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun LongArray.filter(predicate: (Long) ->
Boolean): List<Long> {\n return filterTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\npublic inline fun FloatArray.filter(predicate: (Float) -> Boolean): List<Float> {\n return
filterTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun
DoubleArray.filter(predicate: (Double) -> Boolean): List<Double> {\n return
filterTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun
BooleanArray.filter(predicate: (Boolean) -> Boolean): List<Boolean> {\n return filterTo(ArrayList<Boolean>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun CharArray.filter(predicate: (Char) ->
Boolean): List<Char> {\n return filterTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element
and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun <T> Array<out
T>.filterIndexed(predicate:
(index: Int, T) -> Boolean): List<T> {\n return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns
a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index
of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n *
@sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun
ByteArray.filterIndexed(predicate: (index: Int, Byte) -> Boolean): List<Byte> {\n return
filterIndexedTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun ShortArray.filterIndexed(predicate:
(index: Int, Short) -> Boolean): List<Short>
{\n return filterIndexedTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element
itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun IntArray.filterIndexed(predicate:
(index: Int, Int) -> Boolean): List<Int> {\n return filterIndexedTo(ArrayList<Int>(), predicate)\n}\n\n/**\n *
Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n *
\n * @sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun
LongArray.filterIndexed(predicate: (index: Int, Long) -> Boolean): List<Long> {\n return

```

```

filterIndexedTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun FloatArray.filterIndexed(predicate: (index: Int, Float) -> Boolean): List<Float> {\n return
filterIndexedTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun DoubleArray.filterIndexed(predicate:
(index: Int, Double) -> Boolean): List<Double> {\n return filterIndexedTo(ArrayList<Double>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun
BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {\n return
filterIndexedTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching
the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun CharArray.filterIndexed(predicate:
(index: Int, Char) -> Boolean): List<Char> {\n return filterIndexedTo(ArrayList<Char>(), predicate)\n}\n\n/**\n *
Appends all elements matching
the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element
and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>>
Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n forEachIndexed {
index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*/\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate:
(index: Int, Byte) -> Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element))
destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Short>>
ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C {\n forEachIndexed {
index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic
inline fun <C : MutableCollection<in Int>> IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) ->
Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element))
destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {\n forEachIndexed {

```

```

index, element ->\n if (predicate(index, element)) destination.add(element)\n } \n return
destination\n}\n\n/**\n
 * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function
that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the
element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C :
MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean):
C {\n forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n} \n
return destination\n}\n\n/**\n
 * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterIndexedTo(destination:
C, predicate: (index: Int, Double) -> Boolean): C {\n forEachIndexed { index, element ->\n if
(predicate(index, element)) destination.add(element)\n }\n} \n return destination\n}\n\n/**\n
 * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an
element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in
Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {\n
forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n} \n return
destination\n}\n\n/**\n
 * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and
the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Char>>
CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n forEachIndexed {
index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n} \n return
destination\n}\n\n/**\n
 * Returns a list containing all elements that are instances of specified type parameter R.\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R>
Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n return
filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n
 * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C
: MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {\n for (element in this) if (element is
R) destination.add(element)\n} \n return destination\n}\n\n/**\n
 * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T>
Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {\n return filterNotTo(ArrayList<T>(),
predicate)\n}\n\n/**\n
 * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filterNot(predicate: (Byte) ->
Boolean): List<Byte> {\n return filterNotTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n
 * Returns a list containing
all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*/\npublic inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short>
{\n return filterNotTo(ArrayList<Short>(), predicate)\n}\n\n/**\n
 * Returns a list containing all elements not
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline
fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {\n return filterNotTo(ArrayList<Int>(),
predicate)\n}\n\n/**\n
 * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun LongArray.filterNot(predicate: (Long) ->
Boolean): List<Long> {\n return filterNotTo(ArrayList<Long>(), predicate)\n}\n\n/**\n
 * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filterNot(predicate: (Float) ->
Boolean): List<Float> {\n return filterNotTo(ArrayList<Float>(), predicate)\n}\n\n/**\n
 * Returns a list

```



containing all elements

```
not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *^\npublic inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {\n return filterNotTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *^\npublic inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {\n return filterNotTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *^\npublic inline fun CharArray.filterNot(predicate: (Char) -> Boolean): List<Char> {\n return filterNotTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing all elements that are not `null`.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n *^\npublic fun <T : Any> Array<out T?>.filterNotNull(): List<T> {\n return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n *^\npublic fun <C : MutableCollection<in T>, T : Any> Array<out T?>.filterNotNullTo(destination: C): C {\n for (element in this) if (element != null) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample
```



```

copyOfRange(indices.start, indices.endInclusive + 1).asList()\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun\nFloatArray.slice(indices: IntRange): List<Float> {\n if (indices.isEmpty()) return listOf()\n return\n copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at\n indices in the specified [indices] range.\n */\npublic fun DoubleArray.slice(indices: IntRange): List<Double> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +\n 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun BooleanArray.slice(indices: IntRange): List<Boolean> {\n if (indices.isEmpty()) return listOf()\n return\n copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at\n indices in the specified [indices] range.\n */\npublic fun CharArray.slice(indices: IntRange): List<Char> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +\n 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun <T> Array<out\n T>.slice(indices: Iterable<Int>): List<T> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0)\n return emptyList()\n val list = ArrayList<T>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun\nByteArray.slice(indices: Iterable<Int>): List<Byte> {\n val size = indices.collectionSizeOrDefault(10)\n if (size\n == 0) return emptyList()\n val list = ArrayList<Byte>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun\nShortArray.slice(indices: Iterable<Int>): List<Short> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0)\n return emptyList()\n val list = ArrayList<Short>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun\nIntArray.slice(indices: Iterable<Int>): List<Int> {\n val size =\n indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Int>(size)\n for\n (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at\n specified [indices].\n */\npublic fun LongArray.slice(indices: Iterable<Int>): List<Long> {\n val size =\n indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Long>(size)\n for\n (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing\n elements at specified [indices].\n */\npublic fun FloatArray.slice(indices:\n Iterable<Int>): List<Float> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return\n emptyList()\n val list = ArrayList<Float>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun\nDoubleArray.slice(indices: Iterable<Int>): List<Double> {\n val size = indices.collectionSizeOrDefault(10)\n if\n (size == 0) return emptyList()\n val list = ArrayList<Double>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun BooleanArray.slice(indices: Iterable<Int>): List<Boolean> {\n val size =\n indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Boolean>(size)\n for\n (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun CharArray.slice(indices: Iterable<Int>):\n List<Char> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =\n ArrayList<Char>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun <T>\n Array<T>.sliceArray(indices: Collection<Int>): Array<T> {\n val result = arrayOfNulls(this, indices.size)\n var\n targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return\n result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun\n ByteArray.sliceArray(indices: Collection<Int>): ByteArray {\n val result = ByteArray(indices.size)\n var\n targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++]\n = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at\n specified [indices].\n */\npublic fun ShortArray.sliceArray(indices: Collection<Int>): ShortArray {\n val result =

```

```

ShortArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] =
this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified
[indices].\n */\npublic fun IntArray.sliceArray(indices: Collection<Int>): IntArray {\n val result =
IntArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] =
this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified
[indices].\n */\npublic fun LongArray.sliceArray(indices: Collection<Int>): LongArray {\n val result =
LongArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices)
{\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing
elements of this array at specified [indices].\n */\npublic fun FloatArray.sliceArray(indices: Collection<Int>):
FloatArray {\n val result = FloatArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n
result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements
of this array at specified [indices].\n */\npublic fun DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray
{\n val result = DoubleArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n
result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements
of this array at specified [indices].\n */\npublic fun BooleanArray.sliceArray(indices: Collection<Int>):
BooleanArray {\n val result = BooleanArray(indices.size)\n
var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n
return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
CharArray.sliceArray(indices: Collection<Int>): CharArray {\n val result = CharArray(indices.size)\n var
targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return
result\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
<T> Array<T>.sliceArray(indices: IntRange): Array<T> {\n if (indices.isEmpty()) return copyOfRange(0, 0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n */\npublic fun ByteArray.sliceArray(indices: IntRange): ByteArray {\n if
(indices.isEmpty()) return ByteArray(0)\n return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices
in the specified [indices] range.\n */\npublic fun ShortArray.sliceArray(indices: IntRange): ShortArray {\n if
(indices.isEmpty()) return ShortArray(0)\n return copyOfRange(indices.start, indices.endInclusive +
1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
IntArray.sliceArray(indices: IntRange): IntArray {\n if (indices.isEmpty()) return IntArray(0)\n return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices
in the specified [indices] range.\n */\npublic fun LongArray.sliceArray(indices: IntRange): LongArray {\n if
(indices.isEmpty()) return LongArray(0)\n return copyOfRange(indices.start, indices.endInclusive +
1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic
fun FloatArray.sliceArray(indices: IntRange): FloatArray {\n if (indices.isEmpty()) return FloatArray(0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n */\npublic fun DoubleArray.sliceArray(indices: IntRange): DoubleArray
{\n if (indices.isEmpty()) return DoubleArray(0)\n return copyOfRange(indices.start, indices.endInclusive +
1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
BooleanArray.sliceArray(indices: IntRange): BooleanArray {\n if (indices.isEmpty()) return BooleanArray(0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n */\npublic fun CharArray.sliceArray(indices: IntRange): CharArray {\n
if (indices.isEmpty()) return CharArray(0)\n return copyOfRange(indices.start, indices.endInclusive
+ 1)\n}\n\n/**\n * Returns a list containing first [n] elements.\n */\n * @throws IllegalArgumentException if [n] is
negative.\n */\n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out
T>.take(n: Int): List<T> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n if (n == 0)
return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list
= ArrayList<T>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n}

```

```

return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
ByteArray.take(n: Int): List<Byte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if
(n == 0) return emptyList()\n if (n
 >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Byte>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.take(n: Int):
List<Short> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return
emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list =
ArrayList<Short>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n
return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.take(n: Int): List<Int> {\n
 require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n if (n == 0) return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n
var count = 0\n val list = ArrayList<Int>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n
 break\n }\n return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun LongArray.take(n: Int): List<Long> {\n
 require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Long>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun FloatArray.take(n: Int): List<Float> {\n
 require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Float>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun DoubleArray.take(n: Int): List<Double> {\n
 require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count =
0\n val list = ArrayList<Double>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n
 break\n }\n return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun BooleanArray.take(n: Int): List<Boolean> {\n
 require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Boolean>(n)\n
for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException
if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
CharArray.take(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if
(n == 0) return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n
val list = ArrayList<Char>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n break\n
 }\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val

```



```

samples.collections.Collections.Transformations.take\n *^\npublic inline fun <T> Array<out
T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if
(!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun ByteArray.takeLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun ShortArray.takeLastWhile(predicate: (Short) -> Boolean): List<Short> {\n for (index in lastIndex
downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun
IntArray.takeLastWhile(predicate: (Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if
(!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun LongArray.takeLastWhile(predicate:
(Long) -> Boolean): List<Long> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {\n for (index in
lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n }\n return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun DoubleArray.takeLastWhile(predicate:
(Double) -> Boolean): List<Double> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
return drop(index + 1)\n }\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n for (index in
lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
return drop(index + 1)\n }\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Char> {\n for (index in lastIndex
downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n }\n return
toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun <T> Array<out
T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n val list = ArrayList<T>()\n for (item in this) {\n if
(!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun ByteArray.takeWhile(predicate: (Byte) -> Boolean): List<Byte> {\n val list =
ArrayList<Byte>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return
list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun ShortArray.takeWhile(predicate:
(Short) -> Boolean): List<Short> {\n val list = ArrayList<Short>()\n for (item in this) {\n if
(!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun IntArray.takeWhile(predicate: (Int) -> Boolean): List<Int> {\n val list = ArrayList<Int>()\n
for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list
containing first elements satisfying the given [predicate].\n * \n * @sample

```

```

samples.collections.Collections.Transformations.take\n *^\npublic inline fun LongArray.takeWhile(predicate:
(Long) -> Boolean): List<Long> {\n val list = ArrayList<Long>()\n for (item in this) {\n if
(!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*^\npublic inline fun FloatArray.takeWhile(predicate: (Float) -> Boolean): List<Float> {\n val list =
ArrayList<Float>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n
* \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic inline fun
DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {\n val list = ArrayList<Double>()\n
for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n
* Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun BooleanArray.takeWhile(predicate:
(Boolean) -> Boolean): List<Boolean> {\n val list = ArrayList<Boolean>()\n for (item in this) {\n if
(!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*^\npublic inline fun CharArray.takeWhile(predicate: (Char) -> Boolean): List<Char> {\n val
list = ArrayList<Char>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n
}\n return list\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun <T> Array<T>.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun ByteArray.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun ShortArray.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun IntArray.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun LongArray.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun FloatArray.reverse(): Unit {\n val
midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint)
{\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun DoubleArray.reverse():
Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in
0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun
BooleanArray.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex =
lastIndex\n for (index in 0..midPoint) {\n val
tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n
 }\n}\n\n/**\n * Reverses elements in the array in-place.\n *^\npublic fun CharArray.reverse(): Unit {\n val
midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint)
{\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
 reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param

```



```

fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to
reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun <T> Array<T>.reverse(fromIndex: Int, toIndex:
Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex +
toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex
until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] =
tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n *
@param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive)
to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun ByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex +
toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex
until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] =
tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n *
@param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive)
to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if
(fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex
until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] =
tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n *
@param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive)
to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if
(fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n
 val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex]
= tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n *
@param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range
(exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n *\n@SinceKotlin("1.4")\npublic fun LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if
(fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n
 val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range
in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of
the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n *\n@SinceKotlin("1.4")\npublic fun FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if
(fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n
 val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n
reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range
in-place.\n * \n * @param fromIndex the start

```

of the range (inclusive) to reverse.\n \* @param toIndex the end of the range (exclusive) to reverse.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \*/\n@SinceKotlin("1.4")\npublic fun DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/\*\*\n \* Reverses elements of the array in the specified range in-place.\n \* \n \* @param fromIndex the start of the range (inclusive) to reverse.\n \* @param toIndex the end of the range (exclusive) to reverse.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \*/\n@SinceKotlin("1.4")\npublic fun BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/\*\*\n \* Reverses elements of the array in the specified range in-place.\n \* \n \* @param fromIndex the start of the range (inclusive) to reverse.\n \* @param toIndex the end of the range (exclusive) to reverse.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \*/\n@SinceKotlin("1.4")\npublic fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun <T> Array<out T>.reversed(): List<T> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun ByteArray.reversed(): List<Byte> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun ShortArray.reversed(): List<Short> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun IntArray.reversed(): List<Int> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun LongArray.reversed(): List<Long> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun FloatArray.reversed(): List<Float> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun DoubleArray.reversed(): List<Double> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun BooleanArray.reversed(): List<Boolean> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns a list with elements in reversed order.\n \*/\npublic fun CharArray.reversed(): List<Char> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/\*\*\n \* Returns an array with elements of this array in reversed order.\n \*/\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n if (isEmpty()) return this\n val result = arrayOfNulls(this, size)\n val lastIndex = lastIndexOf()\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/\*\*\n \* Returns an array with elements of this array in reversed order.\n \*/\npublic fun ByteArray.reversedArray(): ByteArray {\n if (isEmpty()) return this\n val result = ByteArray(size)\n val lastIndex = lastIndexOf()\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/\*\*\n \* Returns an array with elements of this array in reversed order.\n \*/\npublic fun

```

ShortArray.reversedArray(): ShortArray {
 if (isEmpty()) return this
 val result = ShortArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun IntArray.reversedArray(): IntArray {
 if (isEmpty()) return this
 val result = IntArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun LongArray.reversedArray(): LongArray {
 if (isEmpty()) return this
 val result = LongArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun FloatArray.reversedArray(): FloatArray {
 if (isEmpty()) return this
 val result = FloatArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun DoubleArray.reversedArray(): DoubleArray {
 if (isEmpty()) return this
 val result = DoubleArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun BooleanArray.reversedArray(): BooleanArray {
 if (isEmpty()) return this
 val result = BooleanArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Returns an array with elements of this array in reversed order.

public fun CharArray.reversedArray(): CharArray {
 if (isEmpty()) return this
 val result = CharArray(size)
 val lastIndex = lastIndex
 for (i in 0..lastIndex)
 result[lastIndex - i] = this[i]
 return result
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun <T> Array<T>.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun ByteArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun ShortArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun IntArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun LongArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun FloatArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun DoubleArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun BooleanArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place.

@SinceKotlin("1.4")
public fun CharArray.shuffle(): Unit {
 shuffle(Random)
}

Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.

See: https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm

@SinceKotlin("1.4")
public fun <T> Array<T>.shuffle(random: Random): Unit {
 for (i in lastIndex downTo 1) {
 val j = random.nextInt(i + 1)
 val copy = this[i]
 this[i] = this[j]
 this[j] = copy
 }
}

Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.

See: https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm

@SinceKotlin("1.4")
public fun ByteArray.shuffle(random: Random): Unit {
 for (i in lastIndex downTo 1) {
 val j = random.nextInt(i + 1)
 val copy = this[i]
 this[i] = this[j]
 this[j] = copy
 }
}

Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.

See: https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm

@SinceKotlin("1.4")
public fun ShortArray.shuffle(random: Random): Unit {
 for (i in lastIndex downTo 1) {
 val j = random.nextInt(i + 1)
 val copy = this[i]
 this[i] = this[j]
 this[j] = copy
 }
}

Randomly shuffles

```

elements in this array in-place using the specified [random] instance as the source of randomness.\n \* \n \* See:  
[https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\\_shuffle#The\\_modern\\_algorithm](https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm)\n

```

*\n@SinceKotlin("1.4")\npublic fun IntArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1)\n {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the\n * source of randomness.\n * \n * See:

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.shuffle(random: Random): Unit {\n for\n (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]\n * instance as the source of randomness.\n * \n * See:

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo\n 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the\n * source of randomness.\n * \n * See:

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.shuffle(random: Random): Unit {\n for (i in lastIndex\n downTo 1) {\n val j = random.nextInt(i\n + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles\n * elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See:

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n
*\n@SinceKotlin("1.4")\npublic fun BooleanArray.shuffle(random: Random): Unit {\n for (i in lastIndex\n downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the\n * source of randomness.\n * \n * See:

https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n
*\n@SinceKotlin("1.4")\npublic fun CharArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo\n 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Sorts elements in the array in-place according to natural sort order of the\n * value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve\n * their order relative to each other after sorting.\n */\n\npublic inline fun <T, R : Comparable<R>> Array<out\n * T>.sortBy(crossinline selector: (T) -> R?): Unit {\n if (size > 1) sortWith(compareBy(selector))\n}\n\n/**\n * Sorts elements in the array in-place descending according to natural sort order of the value returned by specified\n * [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each\n * other after sorting.\n */\n\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortByDescending(crossinline\n * selector: (T) -> R?): Unit {\n if (size > 1) sortWith(compareByDescending(selector))\n}\n\n/**\n * Sorts elements\n * in the array in-place descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n\npublic fun <T : Comparable<T>> Array<out T>.sortDescending(): Unit {\n sortWith(reverseOrder())\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural\n * sort order.\n */\n\npublic fun ByteArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\n\npublic fun\n * ShortArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts\n * elements in the array in-place descending according to their natural sort order.\n */\n\npublic fun\n * IntArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements\n * in the array in-place descending according to their natural sort order.\n */\n\npublic fun LongArray.sortDescending():

```

```

Unit {n if (size > 1) {n sort()n reverse()n }n}n/n/**n * Sorts elements in the array in-place
descending according to their natural sort order.n */npublic fun FloatArray.sortDescending(): Unit {n if (size >
1) {n sort()n reverse()n }n}n/n/**n * Sorts elements in the array in-place descending according to
their natural sort order.n */npublic fun DoubleArray.sortDescending(): Unit {n if (size > 1) {n sort()n
reverse()n }n}n/n/**n * Sorts elements in the array in-place descending according to their natural sort order.n
*/npublic fun CharArray.sortDescending(): Unit {n if (size > 1) {n sort()n reverse()n }n}n/n/**n *
Returns a list of all elements sorted according to their natural sort order.n * n * The sort is _stable_. It means that
equal elements preserve their order relative to each other after sorting.n */npublic fun <T : Comparable<T>>
Array<out T>.sorted():
List<T> {n return sortedArray().asList()n}n/n/**n * Returns a list of all elements sorted according to their
natural sort order.n */npublic fun ByteArray.sorted(): List<Byte> {n return toTypedArray().apply { sort()
}.asList()n}n/n/**n * Returns a list of all elements sorted according to their natural sort order.n */npublic fun
ShortArray.sorted(): List<Short> {n return toTypedArray().apply { sort() }.asList()n}n/n/**n * Returns a list of
all elements sorted according to their natural sort order.n */npublic fun IntArray.sorted(): List<Int> {n return
toTypedArray().apply { sort() }.asList()n}n/n/**n * Returns a list of all elements sorted according to their natural
sort order.n */npublic fun LongArray.sorted(): List<Long> {n return toTypedArray().apply { sort()
}.asList()n}n/n/**n * Returns a list of all elements sorted according to their natural sort order.n */npublic fun
FloatArray.sorted(): List<Float> {n return toTypedArray().apply
{ sort() }.asList()n}n/n/**n * Returns a list of all elements sorted according to their natural sort order.n
*/npublic fun DoubleArray.sorted(): List<Double> {n return toTypedArray().apply { sort() }.asList()n}n/n/**n
* Returns a list of all elements sorted according to their natural sort order.n */npublic fun CharArray.sorted():
List<Char> {n return toTypedArray().apply { sort() }.asList()n}n/n/**n * Returns an array with all elements of
this array sorted according to their natural sort order.n * n * The sort is _stable_. It means that equal elements
preserve their order relative to each other after sorting.n */npublic fun <T : Comparable<T>>
Array<T>.sortedArray(): Array<T> {n if (isEmpty()) return thisn return this.copyOf().apply { sort()
}n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun ByteArray.sortedArray(): ByteArray {n if (isEmpty()) return thisn return this.copyOf().apply
{ sort() }n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun ShortArray.sortedArray(): ShortArray {n if (isEmpty()) return thisn return this.copyOf().apply {
sort() }n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun IntArray.sortedArray(): IntArray {n if (isEmpty()) return thisn return this.copyOf().apply { sort()
}n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun LongArray.sortedArray(): LongArray {n if (isEmpty()) return thisn return this.copyOf().apply {
sort() }n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun FloatArray.sortedArray(): FloatArray {n if (isEmpty()) return thisn return this.copyOf().apply {
sort() }n}n/n/**n * Returns
an array with all elements of this array sorted according to their natural sort order.n */npublic fun
DoubleArray.sortedArray(): DoubleArray {n if (isEmpty()) return thisn return this.copyOf().apply { sort()
}n}n/n/**n * Returns an array with all elements of this array sorted according to their natural sort order.n
*/npublic fun CharArray.sortedArray(): CharArray {n if (isEmpty()) return thisn return this.copyOf().apply {
sort() }n}n/n/**n * Returns an array with all elements of this array sorted descending according to their natural
sort order.n * n * The sort is _stable_. It means that equal elements preserve their order relative to each other after
sorting.n */npublic fun <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {n if (isEmpty())
return thisn return this.copyOf().apply { sortWith(reverseOrder()) }n}n/n/**n * Returns an array with all
elements of this array sorted descending according to their natural sort order.n
*/npublic fun ByteArray.sortedArrayDescending(): ByteArray {n if (isEmpty()) return thisn return
this.copyOf().apply { sortDescending() }n}n/n/**n * Returns an array with all elements of this array sorted
descending according to their natural sort order.n */npublic fun ShortArray.sortedArrayDescending(): ShortArray

```

```

{\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array
with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun
IntArray.sortedArrayDescending(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply {
sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to
their natural sort order.\n */\npublic fun LongArray.sortedArrayDescending(): LongArray {\n if (isEmpty()) return
this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements
of this array sorted descending according to their natural sort order.\n */\npublic fun
FloatArray.sortedArrayDescending(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply {
sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to
their natural sort order.\n */\npublic fun DoubleArray.sortedArrayDescending(): DoubleArray {\n if (isEmpty())
return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of
this array sorted descending according to their natural sort order.\n */\npublic fun
CharArray.sortedArrayDescending(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply {
sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according the specified
[comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other
after sorting.\n */\npublic
fun <T> Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T> {\n if (isEmpty()) return
this\n return this.copyOf().apply { sortWith(comparator) }\n}\n\n/**\n * Returns a list of all elements sorted
according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>> Array<out
T>.sortedBy(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n *
Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector]
function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R :
Comparable<R>> ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R :
Comparable<R>> ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>>
IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>>
LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>>
FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>>
DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
of the value returned by specified [selector] function.\n * \n * @sample

```

`samples.collections.Collections.Sorting.sortedBy` \n \* \n public inline fun <R : Comparable<R>>  
`BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean>` { \n return  
`sortedWith(compareBy(selector))` \n } \n \n /\*\* \n \* Returns a list of all elements sorted according to natural sort order  
of the value returned by specified [selector] function. \n \* \n \* @sample  
`samples.collections.Collections.Sorting.sortedBy` \n \* \n public inline fun <R : Comparable<R>>  
`CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char>` { \n return  
`sortedWith(compareBy(selector))` \n } \n \n /\*\* \n \* Returns a list of all elements sorted descending according to natural  
sort order of the value returned by specified [selector] function. \n \* \n \* The sort is `_stable_`. It means that equal  
elements preserve their order relative to each other after sorting. \n \* \n public inline fun <T, R : Comparable<R>>  
`Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T>` { \n return  
`sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n  
\* Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `ByteArray.sortedByDescending(crossinline  
selector: (Byte) -> R?): List<Byte>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \* Returns  
a list of all elements sorted descending according to natural sort order of the value returned by specified [selector]  
function. \n \* \n public inline fun <R : Comparable<R>> `ShortArray.sortedByDescending(crossinline selector:  
(Short) -> R?): List<Short>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \* Returns a list of  
all elements sorted descending according to natural sort order of the value returned by specified [selector]  
function. \n \* \n public inline fun <R : Comparable<R>> `IntArray.sortedByDescending(crossinline selector: (Int) ->  
R?): List<Int>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n  
\* Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `LongArray.sortedByDescending(crossinline  
selector: (Long) -> R?): List<Long>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \*  
Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `FloatArray.sortedByDescending(crossinline  
selector: (Float) -> R?): List<Float>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \*  
Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `DoubleArray.sortedByDescending(crossinline  
selector: (Double) -> R?): List<Double>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n  
\* Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `BooleanArray.sortedByDescending(crossinline  
selector: (Boolean) -> R?): List<Boolean>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \*  
Returns a list of all elements sorted descending according to natural sort order of the value returned by specified  
[selector] function. \n \* \n public inline fun <R : Comparable<R>> `CharArray.sortedByDescending(crossinline  
selector: (Char) -> R?): List<Char>` { \n return `sortedWith(compareByDescending(selector))` \n } \n \n /\*\* \n \* Returns  
a list of all elements sorted descending according to their natural sort order. \n \* \n \* The sort is `_stable_`. It means  
that equal elements preserve their order relative to each other after sorting. \n \* \n public fun <T : Comparable<T>>  
`Array<out T>.sortedDescending(): List<T>` { \n return `sortedWith(reverseOrder())` \n } \n \n /\*\* \n  
\* Returns a list of all elements sorted descending according to their natural sort order. \n \* \n public fun  
`ByteArray.sortedDescending(): List<Byte>` { \n return `copyOf().apply { sort() }.reversed()` \n } \n \n /\*\* \n \* Returns a  
list of all elements sorted descending according to their natural sort order. \n \* \n public fun  
`ShortArray.sortedDescending(): List<Short>` { \n return `copyOf().apply { sort() }.reversed()` \n } \n \n /\*\* \n \* Returns  
a list of all elements sorted descending according to their natural sort order. \n \* \n public fun  
`IntArray.sortedDescending(): List<Int>` { \n return `copyOf().apply { sort() }.reversed()` \n } \n \n /\*\* \n \* Returns a list  
of all elements sorted descending according to their natural sort order. \n \* \n public fun  
`LongArray.sortedDescending(): List<Long>` { \n return `copyOf().apply { sort() }.reversed()` \n } \n \n /\*\* \n \* Returns  
a list of all elements sorted descending according to their natural sort order. \n \* \n public fun  
`FloatArray.sortedDescending():`

`List<Float> { \n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedDescending(): List<Double> { \n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedDescending(): List<Char> { \n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedWith(comparator: Comparator<in T>): List<T> { \n return sortedArrayWith(comparator).asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> { \n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun <T> Array<out T>.asList(): List<T>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun ByteArray.asList(): List<Byte>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun ShortArray.asList(): List<Short>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun IntArray.asList(): List<Int>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun LongArray.asList(): List<Long>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun FloatArray.asList(): List<Float>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun DoubleArray.asList(): List<Double>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun BooleanArray.asList(): List<Boolean>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun CharArray.asList(): List<Char>\n\n/**\n * Returns true` if the two specified arrays are *deeply* equal to one another, \n * i.e. contain the same number of the same elements in the same order.\n */\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean\n\n/**\n * Returns true` if the two specified arrays are *deeply* equal to one another, \n * i.e. contain the same number of the same elements in the same order.\n */\n * \n * The specified arrays are also considered deeply equal if both are `null`.\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n */\n * \n`



\* The elements of other types are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@SinceKotlin("1.4")\npublic expect infix fun <T> Array<out T>?.contentDeepEquals(other: Array<out T>?): Boolean\n\n/\*\*\n \* Returns a hash code based on the contents of this array as if it is [List].\n \* Nested arrays are treated as lists too.\n \* \n \* If any of arrays contains itself on any nesting level the behavior is undefined.\n

\*\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepHashCode(): Int\n\n/\*\*\n \* Returns a hash code based on the contents of this array as if it is [List].\n \* Nested arrays are treated as lists too.\n \* \n \* If any of arrays contains itself on any nesting level the behavior is undefined.\n

\*\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentDeepHashCode(): Int\n\n/\*\*\n \* Returns a string representation of the contents of this array as if it is a [List].\n \* Nested arrays are treated as lists too.\n \* \n \* If any of arrays contains itself on any nesting level that reference\n \* is rendered as `[...]` to prevent recursion.\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n

\*\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of this array as if it is a [List].\n \* Nested arrays are treated as lists too.\n \* \n \* If any of arrays contains itself on any nesting level that reference\n \* is rendered as `[...]` to prevent recursion.\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n

\*\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentDeepToString(): String\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the same elements in the same order.\n \* \n \* The elements are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the same elements in the same order.\n \* \n \* The elements are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the same elements in the same order.\n \* \n \* The elements are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the same elements in the same order.\n \* \n \* \n \* The elements are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun IntArray.contentEquals(other: IntArray): Boolean\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the same elements in the same order.\n \* \n \* \n \* The elements are compared for equality with the [equals][Any.equals] function.\n \* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

\*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun LongArray.contentEquals(other: LongArray): Boolean\n\n/\*\*\n \* Returns `true` if the two specified arrays are \*structurally\* equal to one another,\n \* i.e. contain the same number of the





```

[List].\n *^n@SinceKotlin("1.4")\npublic expect fun LongArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *^n@SinceKotlin("1.4")\npublic expect fun FloatArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *^n@SinceKotlin("1.4")\npublic expect fun DoubleArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *^n@SinceKotlin("1.4")\npublic expect fun BooleanArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *^n@SinceKotlin("1.4")\npublic expect fun CharArray?.contentHashCode(): Int\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun <T> Array<out T>.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun ByteArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun ShortArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun IntArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun LongArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun FloatArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun DoubleArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun BooleanArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun CharArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n *^n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *^n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified

```

array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun ByteArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun ShortArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun IntArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun LongArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun FloatArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun DoubleArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun BooleanArray?.contentToString(): String\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample samples.collections.Arrays.ContentOperations.contentToString\n \* \n \* @SinceKotlin("1.4")\npublic expect fun CharArray?.contentToString(): String\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \* \n \* @SinceKotlin("1.3")\npublic expect fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \* \n \* @SinceKotlin("1.3")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end

(exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \*/\n\n@SinceKotlin("1.3")\npublic expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \*/\n\n@SinceKotlin("1.3")\npublic expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \*/\n\n@SinceKotlin("1.3")\npublic expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \*/\n\n@SinceKotlin("1.3")\npublic expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray\n\n/\*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the

[destination] array.\n \* \n @SinceKotlin("1.3")\n public expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray\n \n \*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \* \n @SinceKotlin("1.3")\n public expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray\n \n \*\*\n \* Copies this array or its subrange into the [destination] array and returns that array.\n \* \n \* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n \* \n \* @param destination the array to copy to.\n \* @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n \* \n @SinceKotlin("1.3")\n public expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n @Suppress("NO\_ACTUAL\_FOR\_EXPECT")\n public expect fun <T> Array<T>.copyOf(): Array<T>\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun ByteArray.copyOf(): ByteArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun ShortArray.copyOf(): ShortArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun IntArray.copyOf(): IntArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun LongArray.copyOf(): LongArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun FloatArray.copyOf(): FloatArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun DoubleArray.copyOf(): DoubleArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun BooleanArray.copyOf(): BooleanArray\n \n \*\*\n \* Returns new array which is a copy of the original array.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n \* \n public expect fun CharArray.copyOf(): CharArray\n \n \*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* \n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the

copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun ByteArray.copyOf(newSize: Int): ByteArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun ShortArray.copyOf(newSize: Int): ShortArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun IntArray.copyOf(newSize: Int): IntArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun LongArray.copyOf(newSize: Int): LongArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun FloatArray.copyOf(newSize: Int): FloatArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun DoubleArray.copyOf(newSize: Int): DoubleArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with `false` values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun BooleanArray.copyOf(newSize: Int): BooleanArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \*/\npublic expect fun CharArray.copyOf(newSize: Int): CharArray\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with `null` values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n \*/\n@Suppress("NO\_ACTUAL\_FOR\_EXPECT")\npublic



expect fun <T> Array<T>.copyOf(newSize: Int): Array<T?>\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\n@Suppress("NO\_ACTUAL\_FOR\_EXPECT")\n\npublic expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray\n\n\*\*\n \* Returns a new array which is a copy of the specified range of the original array.\n \* \n \* @param fromIndex the start of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

\*\n\npublic expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray\n\n\*\*\n \* Fills this array or its subrange with the

```

specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n *
@param toIndex the end of the
range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex]
is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun <T> Array<T>.fill(element:
T, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified
[element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param
toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic
expect fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this
array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int =
size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex
the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex]
is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun IntArray.fill(element: Int, fromIndex: Int
= 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n *
@param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
(exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun LongArray.fill(element: Long, fromIndex:
Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n
* @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
(exclusive) to fill, size of this array by default.\n
* \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size):
Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the
start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of
this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int =
size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n
* \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the
range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex]
is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun BooleanArray.fill(element:
Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified
[element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param
toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic

```

```

expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns the range
of valid indices for the array.\n */\npublic val <T> Array<out T>.indices: IntRange\n get() = IntRange(0,
lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val ByteArray.indices: IntRange\n
 get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val
ShortArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the
array.\n */\npublic val IntArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of
valid indices for the array.\n */\npublic val LongArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n
* Returns the range of valid indices for the array.\n */\npublic val FloatArray.indices: IntRange\n get() =
IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid
indices for the array.\n */\npublic val DoubleArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n
* Returns the range of valid indices for the array.\n */\npublic val BooleanArray.indices: IntRange\n get() =
IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val
CharArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns `true` if the array is empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isEmpty(): Boolean {\n return size ==
0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n
* Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.isEmpty():
Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n
* Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isEmpty():
Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isEmpty(): Boolean {\n return size ==
0\n}\n\n/**\n * Returns `true` if the array is empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n
* Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n
* Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n
* Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isNotEmpty(): Boolean {\n return
!isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline
fun IntArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.isNotEmpty(): Boolean {\n return
!isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline
fun FloatArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not
empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isNotEmpty(): Boolean {\n return
!isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not
empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.isNotEmpty(): Boolean {\n return
!isEmpty()\n}\n\n/**\n * Returns the last valid index for the array.\n */\npublic val <T> Array<out T>.lastIndex:
Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val ByteArray.lastIndex:
Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val ShortArray.lastIndex:
Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val IntArray.lastIndex: Int\n
 get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val LongArray.lastIndex: Int\n
 get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic

```

val FloatArray.lastIndex: Int\n get() = size - 1\n\n/\*\*\n \* Returns the last valid index for the array.\n \*/\npublic  
 val DoubleArray.lastIndex: Int\n get() = size - 1\n\n/\*\*\n \* Returns the last valid index for the array.\n \*/\npublic  
 val BooleanArray.lastIndex: Int\n get() = size - 1\n\n/\*\*\n \* Returns the last valid index for the array.\n \*/\npublic  
 val CharArray.lastIndex: Int\n get() = size - 1\n\n/\*\*\n \* Returns an array containing all elements of the original  
 array and then the given [element].\n \*/\n@Suppress("NO\_ACTUAL\_FOR\_EXPECT")\npublic expect operator  
 fun <T> Array<T>.plus(element: T): Array<T>\n\n/\*\*\n \* Returns an array containing all elements of the original  
 array and then the given [element].\n \*/\npublic expect operator fun ByteArray.plus(element: Byte):  
 ByteArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n \*/\n  
 /\*\*\n public expect operator fun ShortArray.plus(element: Short): ShortArray\n\n/\*\*\n  
 \* Returns an array containing all elements of the original array and then the given [element].\n \*/\npublic expect  
 operator fun IntArray.plus(element: Int): IntArray\n\n/\*\*\n \* Returns an array containing all elements of the original  
 array and then the given [element].\n \*/\npublic expect operator fun LongArray.plus(element: Long):  
 LongArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n \*/\n  
 /\*\*\n public expect operator fun FloatArray.plus(element: Float): FloatArray\n\n/\*\*\n \* Returns an array containing  
 all elements of the original array and then the given [element].\n \*/\npublic expect operator fun  
 DoubleArray.plus(element: Double): DoubleArray\n\n/\*\*\n \* Returns an array containing all elements of the  
 original array and then the given [element].\n \*/\npublic expect operator fun BooleanArray.plus(element: Boolean):  
 BooleanArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then the given  
 [element].\n \*/\n  
 /\*\*\n public expect operator fun CharArray.plus(element: Char): CharArray\n\n/\*\*\n \* Returns an array containing  
 all elements of the original array and then all elements of the given [elements] collection.\n \*/\n  
 /\*\*\n @Suppress("NO\_ACTUAL\_FOR\_EXPECT")\n public expect operator fun <T> Array<T>.plus(elements:  
 Collection<T>): Array<T>\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all  
 elements of the given [elements] collection.\n \*/\npublic expect operator fun ByteArray.plus(elements:  
 Collection<Byte>): ByteArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all  
 elements of the given [elements] collection.\n \*/\npublic expect operator fun ShortArray.plus(elements:  
 Collection<Short>): ShortArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all  
 elements of the given [elements] collection.\n \*/\npublic expect operator fun IntArray.plus(elements:  
 Collection<Int>): IntArray\n\n/\*\*\n  
 \* Returns an array containing all elements of the original array and then all elements of the given [elements]  
 collection.\n \*/\npublic expect operator fun LongArray.plus(elements: Collection<Long>): LongArray\n\n/\*\*\n \*  
 Returns an array containing all elements of the original array and then all elements of the given [elements]  
 collection.\n \*/\npublic expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray\n\n/\*\*\n \*  
 Returns an array containing all elements of the original array and then all elements of the given [elements]  
 collection.\n \*/\npublic expect operator fun DoubleArray.plus(elements: Collection<Double>):  
 DoubleArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the  
 given [elements] collection.\n \*/\npublic expect operator fun BooleanArray.plus(elements: Collection<Boolean>):  
 BooleanArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the  
 given [elements]  
 collection.\n \*/\npublic expect operator fun CharArray.plus(elements: Collection<Char>): CharArray\n\n/\*\*\n \*  
 Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n \*/\n  
 /\*\*\n @Suppress("NO\_ACTUAL\_FOR\_EXPECT")\n public expect operator fun <T> Array<T>.plus(elements:  
 Array<out T>): Array<T>\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all  
 elements of the given [elements] array.\n \*/\npublic expect operator fun ByteArray.plus(elements: ByteArray):  
 ByteArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the  
 given [elements] array.\n \*/\npublic expect operator fun ShortArray.plus(elements: ShortArray):  
 ShortArray\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the  
 given [elements] array.\n \*/\npublic expect operator fun IntArray.plus(elements: IntArray): IntArray\n\n/\*\*\n \*

Returns an array

containing all elements of the original array and then all elements of the given [elements] array.\n \*/\npublic expect operator fun LongArray.plus(elements: LongArray): LongArray\n\n \*/\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n \*/\npublic expect operator fun FloatArray.plus(elements: FloatArray): FloatArray\n\n \*/\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n \*/\npublic expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray\n\n \*/\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n \*/\npublic expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray\n\n \*/\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n \*/\npublic expect operator fun CharArray.plus(elements: CharArray): CharArray\n\n \*/\n \* Returns an array containing all elements of the original array and then the given [element].\n \*/\n\n @Suppress("NO\_ACTUAL\_FOR\_EXPECT")\npublic expect fun <T> Array<T>.plusElement(element: T): Array<T>\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun IntArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun LongArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun ByteArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun ShortArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun DoubleArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun FloatArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun CharArray.sort(): Unit\n\n \*/\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n \*/\npublic expect fun <T : Comparable<T>> Array<out T>.sort(): Unit\n\n \*/\n \* \n \* @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n \* @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \* \n \* @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n \*/\n\n @SinceKotlin("1.4")\npublic expect fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n \*/\n \* \n \* @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n \* @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \* \n \* @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n \*/\n\n @SinceKotlin("1.4")\npublic expect fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n \*/\n \* \n \* @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n \* @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \* \n \* @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n \*/\n\n @SinceKotlin("1.4")\npublic expect fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n \*/\n \* \n \* @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n \* @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n \* @throws IndexOutOfBoundsException if [fromIndex] is

less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex]

[fromIndex]

is greater than [toIndex].\n \* \n \* @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n

```
*\n@SinceKotlin("1.4")\npublic expect fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n *\n@SinceKotlin("1.4")\npublic expect fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n *\n@SinceKotlin("1.4")\npublic expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n *\n@SinceKotlin("1.4")\npublic expect fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n *\n@SinceKotlin("1.4")\npublic expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *\n@SinceKotlin("1.4")\npublic fun <T>.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sortWith(reverseOrder(), fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *\n@SinceKotlin("1.4")\npublic fun ByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified
```

range in-place. \n \* The elements are sorted descending according to their natural sort order. \n \* \n \* @param fromIndex the start of the range (inclusive) to sort. \n \* @param toIndex the end of the range (exclusive) to sort. \n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n

```

*\n@SinceKotlin("1.4")\npublic fun IntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place. \n * The elements are sorted descending according to their natural sort order. \n * \n * @param fromIndex the start of the range (inclusive) to sort. \n * @param toIndex the end of the range (exclusive) to sort. \n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n
```

```

*\n@SinceKotlin("1.4")\npublic fun LongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place. \n * The elements are sorted descending according to their natural sort order. \n * \n * @param fromIndex the start of the range (inclusive) to sort. \n * @param toIndex the end of the range (exclusive) to sort. \n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n
```

```

*\n@SinceKotlin("1.4")\npublic fun FloatArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place. \n * The elements are sorted descending according to their natural sort order. \n * \n * @param fromIndex the start of the range (inclusive) to sort. \n * @param toIndex the end of the range (exclusive) to sort. \n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n
```

```

*\n@SinceKotlin("1.4")\npublic fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place. \n * The elements are sorted descending according to their natural sort order. \n * \n * @param fromIndex the start of the range (inclusive) to sort. \n * @param toIndex the end of the range (exclusive) to sort. \n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n
```

```

*\n@SinceKotlin("1.4")\npublic fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparator]. \n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting. \n
```

```

*\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit\n\n/**\n * Sorts a range in the array in-place with the given [comparator]. \n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting. \n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default. \n * @param toIndex the end of the range (exclusive) to sort, size of this array by default. \n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. \n
```

```

*\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns an array of Boolean containing all of the elements of this generic array. \n
```

```

*\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n return BooleanArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Byte containing all of the elements of this generic array. \n
```

```

*\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n return ByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Char containing all of the elements of this generic array. \n
```

```

*\npublic fun Array<out Char>.toCharArray(): CharArray {\n return CharArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Double containing all of the elements of this generic array. \n
```

```

*\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n return DoubleArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Float containing all of the elements of this generic array. \n
```

```

fun Array<out Float>.toFloatArray(): FloatArray {
 return FloatArray(size) { index -> this[index] }
}

* Returns an array of Int containing all of the elements of this generic array.

public fun Array<out Int>.toIntArray(): IntArray {
 return IntArray(size) { index -> this[index] }
}

* Returns an array of Long containing all of the elements of this generic array.

public fun Array<out Long>.toLongArray(): LongArray {
 return LongArray(size) { index -> this[index] }
}

* Returns an array of Short containing all of the elements of this generic array.

public fun Array<out Short>.toShortArray(): ShortArray {
 return ShortArray(size) { index -> this[index] }
}

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun ByteArray.toTypedArray(): Array<Byte>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun ShortArray.toTypedArray(): Array<Short>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun IntArray.toTypedArray(): Array<Int>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun LongArray.toTypedArray(): Array<Long>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun FloatArray.toTypedArray(): Array<Float>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun DoubleArray.toTypedArray(): Array<Double>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun BooleanArray.toTypedArray(): Array<Boolean>

* Returns a *typed* object array containing all of the elements of this primitive array.

public expect fun CharArray.toTypedArray(): Array<Char>

* Returns a [Map] containing key-value pairs provided by [transform] function
* applied to elements of the given array.
* If any of two pairs would have the same key the last one gets added to the map.
* The returned map preserves the entry iteration order of the original array.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

* Returns a [Map] containing key-value pairs provided by [transform] function
* applied to elements of the given array.
* If any of two pairs would have the same key the last one gets added to the map.
* The returned map preserves the entry iteration order of the original array.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <K, V> ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

* Returns a [Map] containing key-value pairs provided by [transform] function
* applied to elements of the given array.
* If any of two pairs would have the same key the last one gets added to the map.
* The returned map preserves the entry iteration order of the original array.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <K, V> ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

* Returns a [Map] containing key-value pairs provided by [transform] function
* applied to elements of the given array.
* If any of two pairs would have the same key the last one gets added to the map.
* The returned map preserves the entry iteration order of the original array.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <K, V> IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

* Returns a [Map] containing key-value pairs provided by [transform] function
* applied to elements of the given array.
* If any of two pairs would have the same key the last one gets added to the map.
* The returned map preserves the entry iteration order of the original array.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <K, V>

```



```

LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n
 return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-
value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs
would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V>
FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original
array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline
fun <K, V> DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original
array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline
fun <K, V> BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs
provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have
the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the
original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline
fun <K, V> CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n val capacity
= mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <T, K> Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, T>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Byte>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned
from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned
by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of
the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Short>(capacity),

```

```

keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K> IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Int>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K>
LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Long>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same
key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K>
FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Float>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K>
DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {\n val
capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Double>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K>
BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Boolean>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the
given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
*/\npublic inline fun <K>
CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n val capacity =
mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, Char>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by
[keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n

```

```

*

public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V):

Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return

 associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V):

Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return

 associateByTo(LinkedHashMap<K,

 V>(capacity), keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return associateByTo(LinkedHashMap<K, V>(capacity),

 keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return associateByTo(LinkedHashMap<K, V>(capacity),

 keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return associateByTo(LinkedHashMap<K,

 V>(capacity), keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return associateByTo(LinkedHashMap<K, V>(capacity),

 keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.  
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
The returned map preserves the entry iteration order of the original array.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```

```

*

public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V> {

 val capacity = mapCapacity(size).coerceAtLeast(16)

 return associateByTo(LinkedHashMap<K,

 V>(capacity), keySelector, valueTransform)

}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.

[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

\* The returned map preserves the entry iteration order of the original array.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform
*/
public inline
fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V> {
 val capacity = mapCapacity(size).coerceAtLeast(16)
 return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

\* The returned map preserves the entry iteration order of the original array.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform
*/
public inline
fun <K, V> CharArray.associateBy(keySelector:
(Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
 val capacity =
mapCapacity(size).coerceAtLeast(16)
 return associateByTo(LinkedHashMap<K, V>(capacity), keySelector,
valueTransform)
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
*/
public inline
fun <T, K, M : MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) ->
K): M {
 for (element in this) {
 destination.put(keySelector(element), element)
 }
 return
destination
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
*/
public inline
fun <K, M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {
 for
(element in this) {
 destination.put(keySelector(element), element)
 }
 return destination
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
*/
public inline
fun <K, M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K):
M {
 for (element in this) {
 destination.put(keySelector(element), element)
 }
 return
destination
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
*/
public inline
fun <K, M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M
{
 for (element in this) {
 destination.put(keySelector(element), element)
 }
 return
destination
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
*/
public inline
fun <K, M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector:
(Long) -> K): M {
 for (element in this) {
 destination.put(keySelector(element), element)
 }
 return
destination
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the

```

map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n
*/\npublic inline fun <K, M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector:
(Float) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is
provided by the [keySelector] function applied to each element of the given array\n * and value is the element
itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n*/\npublic
inline fun <K, M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector:
(Double) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the
given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by
[keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n*/\npublic inline fun <K, M :
MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {\n
for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n
* Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n
*/\npublic inline fun <K, M : MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector:
(Char) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is
provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to
elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last
one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n*/\npublic
inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -
> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n*/\npublic
inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) ->
K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n*/\npublic
inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) ->
K, valueTransform: (Short) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by
the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n

```

```

*^public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector:
(Int) -> K, valueTransform: (Int) -> V): M {
 for (element in this) {
 destination.put(keySelector(element),
valueTransform(element))
 }
 return destination
}

Populates and returns the [destination] mutable map with key-value pairs,
where key is provided by the [keySelector] function and
and value is provided by the [valueTransform] function applied to elements of the given array.
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

*^public
inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) ->
K, valueTransform: (Long) -> V): M {
 for (element in this)
 destination.put(keySelector(element), valueTransform(element))
 return destination
}

Populates and returns the [destination] mutable map with key-value pairs,
where key is provided by the
[keySelector] function and
and value is provided by the [valueTransform] function applied to elements of the
given array.
If any two elements would have the same key returned by [keySelector] the last one gets added
to the map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

*^public
inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) ->
K, valueTransform: (Float) -> V): M {
 for (element in this) {
 destination.put(keySelector(element),
valueTransform(element))
 }
 return destination
}

Populates and returns the [destination] mutable
map with key-value pairs,
where key is provided by the [keySelector]
function and
and value is provided by the [valueTransform] function applied to elements of the given array.
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

*^public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M,
keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
 for (element in this) {
 destination.put(keySelector(element), valueTransform(element))
 }
 return destination
}

Populates
and returns the [destination] mutable map with key-value pairs,
where key is provided by the [keySelector]
function and
and value is provided by the [valueTransform] function applied to elements of the given array.
If any two elements would have the same key returned by [keySelector] the last one gets added to the
map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

*^public
inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector:
(Boolean) -> K, valueTransform: (Boolean) -> V): M {
 for (element in this) {
 destination.put(keySelector(element), valueTransform(element))
 }
 return destination
}

Populates
and returns the [destination] mutable map with key-value pairs,
where key is provided by the [keySelector]
function and
and value is provided by the [valueTransform] function applied to elements of the given array.
If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

*^public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector:
(Char)
-> K, valueTransform: (Char) -> V): M {
 for (element in this) {
 destination.put(keySelector(element),
valueTransform(element))
 }
 return destination
}

Populates and returns the [destination] mutable
map with key-value pairs
provided by [transform] function applied to each element of the given array.
If any of two pairs would have the same key the last one gets added to the map.
@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

*^public inline fun <T, K, V, M :
MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {
 for
(element in this) {
 destination += transform(element)
 }
 return destination
}

Populates and
returns the [destination] mutable map with key-value pairs
provided by [transform] function applied to each
element of the given array.
If any of two pairs would have the same key the last one gets

```

```

added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform:
(Byte) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n *
provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs
would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M :
MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {\n
for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with
key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n *
If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M :
MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {\n
for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform]
function applied to each element of the given array.\n * \n * If any of two pairs would have the same
key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateTo(destination: M, transform: (Long) ->
Pair<K, V>): M {\n
for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform]
function applied to each element of the given array.\n * \n * If any of two pairs would have the same
key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n *
provided by [transform] function applied to each element of the given array.\n * \n * If any of two
pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform:
(Double) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n
return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value
pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If
any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M :
MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {\n
for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform]
function applied to each element of the given array.\n * \n * If any of two pairs would have the same
key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return
destination\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are
equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\npublic inline fun <K, V> Array<out K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n val result =

```

LinkedHashMap<K,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns  
 a [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the map.\n \* \n \* The  
 returned map preserves the entry iteration order of the original array.\n \* \n \* @sample  
 samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> {\n val result = LinkedHashMap<Byte,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns a  
 [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the  
 map.\n \* \n \* The returned map preserves the entry iteration order of the original array.\n \* \n \* @sample  
 samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> {\n val result = LinkedHashMap<Short,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns a  
 [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the map.\n \* \n \* The  
 returned map preserves the entry iteration order of the original array.\n \* \n \* @sample  
 samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {\n  
 val result = LinkedHashMap<Int, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result,  
 valueSelector)\n}\n\n/\*\*\n \* Returns a [Map] where keys are elements from the given array and values are\n \*  
 produced by the [valueSelector] function applied to each element.\n \* \n \* If any two elements are equal, the last one  
 gets added to the map.\n \* \n \* The returned map preserves the entry iteration order of the original array.\n \* \n \*  
 @sample samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {\n val result = LinkedHashMap<Long,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns a  
 [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal,  
 the last one gets added to the map.\n \* \n \* The returned map preserves the entry iteration order of the original  
 array.\n \* \n \* @sample samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {\n val result = LinkedHashMap<Float,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns a  
 [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the map.\n \* \n \* The  
 returned map preserves the entry iteration order of the original array.\n \* \n \* @sample  
 samples.collections.Collections.Transformations.associateWith\n

\*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>  
 DoubleArray.associateWith(valueSelector:  
 (Double) -> V): Map<Double, V> {\n val result = LinkedHashMap<Double,  
 V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Returns a  
 [Map] where keys are elements from the given array and values are\n \* produced by the [valueSelector] function  
 applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the map.\n \* \n \* The  
 returned map preserves the entry iteration order of the original array.\n \* \n \* @sample



```

samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V> {\n val result =
LinkedHashMap<Boolean, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to
each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map
preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
CharArray.associateWith(valueSelector: (Char) -> V): Map<Char, V> {\n val result = LinkedHashMap<Char,
V>(mapCapacity(size).coerceAtMost(128)).coerceAtLeast(16))\n return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\npublic
inline
fun <K, V, M : MutableMap<in K, in V>> Array<out K>.associateWithTo(destination: M, valueSelector: (K) ->
V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Byte, in V>>
ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination]
mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value
is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one
overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Short, in V>>
ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Int, in V>>
IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n *
@sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {\n for (element in this) {\n

```

```

destination.put(element, valueSelector(element))\n } \n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the
element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements
are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n } \n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector]
function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the
map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n } \n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline
fun <V, M : MutableMap<in Boolean, in V>> BooleanArray.associateWithTo(destination: M, valueSelector:
(Boolean) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n } \n
return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n
* @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Char, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n } \n return destination\n}\n\n/**\n * Appends all elements to
the given
[destination] collection.\n *\npublic fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination:
C): C {\n for (item in this) {\n destination.add(item)\n } \n return destination\n}\n\n/**\n * Appends all
elements to the given [destination] collection.\n *\npublic fun <C : MutableCollection<in Byte>>
ByteArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n } \n return
destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n *\npublic fun <C :
MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {\n for (item in this) {\n
destination.add(item)\n } \n return destination\n}\n\n/**\n * Appends all elements to the given [destination]
collection.\n *\npublic fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {\n for
(item in this) {\n destination.add(item)\n } \n return destination\n}\n\n/**\n * Appends
all elements to the given [destination] collection.\n *\npublic fun <C : MutableCollection<in Long>>
LongArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n } \n return
destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n *\npublic fun <C :
MutableCollection<in Float>> FloatArray.toCollection(destination: C): C {\n for (item in this) {\n
destination.add(item)\n } \n return destination\n}\n\n/**\n * Appends all elements to the given [destination]
collection.\n *\npublic fun <C : MutableCollection<in Double>> DoubleArray.toCollection(destination: C): C {\n
for (item in this) {\n destination.add(item)\n } \n return destination\n}\n\n/**\n * Appends all elements to the
given [destination] collection.\n *\npublic fun <C : MutableCollection<in Boolean>>
BooleanArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n } \n
return destination\n}\n\n

```



```

array.\n */\npublic fun LongArray.toMutableList(): MutableList<Long>
{\n val list = ArrayList<Long>(size)\n for (item in this) list.add(item)\n return list}\n\n/**\n * Returns a
new [MutableList] filled with all elements of this array.\n */\npublic fun FloatArray.toMutableList():
MutableList<Float> {\n val list = ArrayList<Float>(size)\n for (item in this) list.add(item)\n return
list}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun
DoubleArray.toMutableList(): MutableList<Double> {\n val list = ArrayList<Double>(size)\n for (item in this)
list.add(item)\n return list}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n
*/\npublic fun BooleanArray.toMutableList(): MutableList<Boolean> {\n val list = ArrayList<Boolean>(size)\n
for (item in this) list.add(item)\n return list}\n\n/**\n * Returns a new [MutableList] filled with all elements of
this array.\n */\npublic fun CharArray.toMutableList(): MutableList<Char> {\n val
list = ArrayList<Char>(size)\n for (item in this) list.add(item)\n return list}\n\n/**\n * Returns a [Set] of all
elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun <T>
Array<out T>.toSet(): Set<T> {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else -
-> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n *
The returned set preserves the element iteration order of the original array.\n */\npublic fun ByteArray.toSet():
Set<Byte> {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else ->
toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n *
The returned set preserves the element iteration order of the original array.\n */\npublic fun ShortArray.toSet():
Set<Short> {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else -> toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n
*/\npublic fun IntArray.toSet(): Set<Int> {\n return when (size) {\n 0 -> emptySet()\n 1 ->
setOf(this[0])\n else -> toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a
[Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic
fun LongArray.toSet(): Set<Long> {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n
else -> toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all
elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun
FloatArray.toSet(): Set<Float> {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else -> toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the
original array.\n */\npublic fun DoubleArray.toSet(): Set<Double> {\n return when (size) {\n 0 ->
emptySet()\n 1 -> setOf(this[0])\n else -> toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the
original array.\n */\npublic fun BooleanArray.toSet(): Set<Boolean> {\n return when (size) {\n 0 ->
emptySet()\n 1 -> setOf(this[0])\n else -> toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the
original array.\n */\npublic fun CharArray.toSet(): Set<Char> {\n
return when (size) {\n 0 -> emptySet()\n 1 -> setOf(this[0])\n else ->
toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n }\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n
*/\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n
* Returns a single list of all elements yielded from results of [transform] function being invoked on each element of
original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun
<R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked

```

```

on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n
*\npublic inline fun <R> ShortArray.flatMap(transform: (Short) -> Iterable<R>): List<R> {\n return
flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of
[transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R> IntArray.flatMap(transform:
(Int) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R>
LongArray.flatMap(transform: (Long) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R>
FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R>
DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R>
BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>):
List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements
yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n *\npublic inline fun <R> CharArray.flatMap(transform:
(Char) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements
yielded from results of [transform] function being invoked on each element\n * and its index in the original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
\n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
\n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {\n return

```

```

flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded
from results of [transform] function being invoked
on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R> Array<out T>.flatMapIndexed(transform:
(index: Int, T) -> Sequence<R>): List<R> {\n return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n *
Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index
in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Byte) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination:
C, transform: (index: Int, Short) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element\n * and its index in the original
array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Int) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n
 destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked
on each element\n * and its index in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Long) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c
inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Float) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index:
Int, Double) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n

```

```

* Appends all elements yielded from results of [transform] function being invoked on each element\n * and its
index in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index:
Int, Boolean) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c
inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Char) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npub
blic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n *\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C,
transform: (T) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C
{\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C
{\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n *\npublic inline fun <R, C : MutableCollection<in R>>
IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {\n for (element in this) {\n val list =
transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements
yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination:
C, transform: (Long) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C
{\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n *\npublic inline fun <R, C : MutableCollection<in R>>
DoubleArray.flatMapTo(destination: C, transform: (Double) -> Iterable<R>): C {\n for (element in this) {\n
val list = transform(element)\n

```



```

destination.addAll(list)\n } return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline
fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) ->
Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n
}\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being
invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked
on each element of original array, to the given [destination].\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequenceTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun
<T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>>
{\n return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups elements of
the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a
map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves
the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {\n return groupByTo(LinkedHashMap<K,
MutableList<Byte>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {\n return groupByTo(LinkedHashMap<K,
MutableList<Int>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a
list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic
inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>

```

```

FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {\n return groupByTo(LinkedHashMap<K,
MutableList<Float>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
BooleanArray.groupBy(keySelector:
(Boolean) -> K): Map<K, List<Boolean>> {\n return groupByTo(LinkedHashMap<K,
MutableList<Boolean>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by
the given [keySelector] function\n * applied to each element and returns a map where each group key is associated
with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of
corresponding values.\n
* \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n *
@sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K,
V> Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V):
Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of
the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns
a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves
the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n
* and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned
map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values

```

returned by the [valueTransform] function applied to each element of the original array\n \* by the key returned by the given [keySelector] function applied to the element\n \* and returns a map where each group key is associated with a list of corresponding values.\n \* \n \* The returned map preserves the entry iteration order of the keys produced from the original array.\n \* \n \* @sample

```

samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic
inline fun <K, V> LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K,
List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of
the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns
a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves
the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a
map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {\n
return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key
returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is
associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the
keys produced from the original array.\n
\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun
<K, V> BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>>>
{\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n *
Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the
key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key
is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the
keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n *
\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
\n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Array<out T>.groupByTo(destination: M,
keySelector: (T) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<T>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
\n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Byte>>> ByteArray.groupByTo(destination: M,
keySelector: (Byte) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Byte>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each

```

```

element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Short>>>> ShortArray.groupByTo(destination: M,
keySelector: (Short) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Short>() }\n list.add(element)\n
 }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given
[keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with
a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,
MutableList<Int>>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K): M {\n for (element in this) {\n
 val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Int>() }\n
list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned
by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key
associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n
* @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M :
MutableMap<in K, MutableList<Long>>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M
{\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<Long>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <K, M : MutableMap<in K, MutableList<Float>>>> FloatArray.groupByTo(destination: M, keySelector: (Float)
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) {
ArrayList<Float>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <K, M : MutableMap<in K, MutableList<Double>>>> DoubleArray.groupByTo(destination: M, keySelector:
(Double) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Double>() }\n list.add(element)\n }\n return destination\n}\n\n/**
 * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n
* \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,
MutableList<Boolean>>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {\n for
(element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<Boolean>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <K, M : MutableMap<in K, MutableList<Char>>>> CharArray.groupByTo(destination: M, keySelector: (Char)
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n
 val list = destination.getOrPut(key) { ArrayList<Char>() }\n list.add(element)\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic

```

```

inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Array<out T>.groupByTo(destination: M,
keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n val key =
keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n
list.add(valueTransform(element))\n }\n return destination}\n\n/**\n * Groups values returned by the
[valueTransform] function
 applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to
the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n *
\n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K,
valueTransform: (Byte) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with
 a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K,
valueTransform: (Short) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic
inline fun <K, V, M : MutableMap<in
K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M
{\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key
associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K,
valueTransform: (Long) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n
 val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic
inline fun <K, V, M : MutableMap<in K, MutableList<V>>> FloatArray.groupByTo(destination: M, keySelector:
(Float) -> K, valueTransform: (Float) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n
 val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n\n/**\n * Groups values returned by the [valueTransform] function
 applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to
the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n *
\n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K,

```

```

valueTransform: (Double) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list
= destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n}\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated
with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K,
valueTransform: (Boolean) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list
= destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n}\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic
inline fun <K,
V, M : MutableMap<in K, MutableList<V>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n}\n/**\n * Creates a [Grouping] source from an array to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to extract a key from each element.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin("1.1")\npublic inline fun <T, K>
Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n return object : Grouping<T,
K> {\n override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n override fun
keyOf(element: T): K = keySelector(element)\n }\n}\n/**\n * Returns a list containing the results of applying
the given
[transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <T, R> Array<out T>.map(transform:
(T) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ByteArray.map(transform: (Byte)
-> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ShortArray.map(transform:
(Short) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing
the results of applying the
given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> IntArray.map(transform: (Int) ->
R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the results
of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> LongArray.map(transform:
(Long) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> FloatArray.map(transform: (Float)
-> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing the
results of applying the
given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> DoubleArray.map(transform:
(Double) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n}\n/**\n * Returns a list containing

```

the results of applying the given [transform] function to each element in the original array.

```

@sample samples.collections.Collections.Transformations.map
public inline fun <R> BooleanArray.map(transform: (Boolean) -> R): List<R> {
 return mapTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element in the original array.
@sample samples.collections.Collections.Transformations.map
public inline fun <R> CharArray.map(transform: (Char) -> R): List<R> {
 return mapTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <T, R> Array<out T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> ByteArray.mapIndexed(transform: (index: Int, Byte) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> ShortArray.mapIndexed(transform: (index: Int, Short) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> IntArray.mapIndexed(transform: (index: Int, Int) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> DoubleArray.mapIndexed(transform: (index: Int, Double) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> BooleanArray.mapIndexed(transform: (index: Int, Boolean) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing the results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <R> CharArray.mapIndexed(transform: (index: Int, Char) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(size), transform)
}
Returns a list containing only the non-null results of applying the given [transform] function to each element and its index in the original array.
@param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
public inline fun <T, R : Any>

```





```

return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original
array\n * and appends the results to the given [destination].\n * @param [transform]
function that takes the index of an element and the element itself\n * and returns the result of the transform applied
to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> CharArray.mapIndexedTo(destination: C,
transform: (index: Int, Char) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Returns a list containing only the
non-null results of applying the given [transform] function\n * to each element in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline fun <T, R : Any>
Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each element in the original array\n * and
appends only the non-null results to the given [destination].\n */\npublic inline fun <T, R : Any, C :
MutableCollection<in R>>
Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n forEach { element ->
transform(element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic
inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapTo(destination: C, transform: (T) -> R): C {\n
for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given
[transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapTo(destination: C, transform: (Byte) -> R):
C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the
given [transform] function to each element of the original array\n * and appends the results
to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
ShortArray.mapTo(destination: C, transform: (Short) -> R): C {\n for (item in this)\n
 destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {\n for (item in this)\n
 destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C :
MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {\n for (item in this)\n
 destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function
to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun
<R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {\n for (item in
this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic
inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {\n
for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given
[transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean)
-> R): C {\n for (item in this)\n destination.add(transform(item))\n
 return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n *
and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>>
CharArray.mapTo(destination: C, transform: (Char) -> R): C {\n for (item in this)\n
 destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each
element of the original array\n * into an [IndexedValue] containing the index of that element and the element
itself.\n */\npublic fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {\n return IndexingIterable {
iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an
[IndexedValue] containing the index of that element and the element itself.\n */\npublic fun ByteArray.withIndex():
Iterable<IndexedValue<Byte>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]

```

that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun LongArray.withIndex():
Iterable<IndexedValue<Long>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun DoubleArray.withIndex():
Iterable<IndexedValue<Double>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>> {
 return IndexingIterable { iterator() }
}

```

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun CharArray.withIndex(): Iterable<IndexedValue<Char>> {
 return IndexingIterable { iterator() }
}

```

Returns a list containing only distinct elements from the given array. Among equal elements of the given array, only the first one will be present in the resulting list. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun <T> Array<out T>.distinct(): List<T> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun ByteArray.distinct(): List<Byte> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in

the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun ShortArray.distinct(): List<Short> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun IntArray.distinct():
List<Int> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun LongArray.distinct(): List<Long> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun FloatArray.distinct(): List<Float> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

public fun DoubleArray.distinct():
List<Double> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy

```

```

BooleanArray.distinct(): List<Boolean> {\n return
 this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n *
\n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun CharArray.distinct():
List<Char> {\n return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only elements from the
given array\n * having distinct keys returned by the given [selector] function.\n * \n * Among elements of the given
array with equal keys, only the first one will be present in the resulting list.\n * The elements in the resulting list are
in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <T, K> Array<out
T>.distinctBy(selector: (T) -> K): List<T> {\n val set = HashSet<K>()\n val list = ArrayList<T>()\n for (e
in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n *
Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector]
function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n *
@sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K>
ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {\n val set = HashSet<K>()\n val list =
ArrayList<Byte>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> ShortArray.distinctBy(selector: (Short) -> K): List<Short> {\n val set =
HashSet<K>()\n val list = ArrayList<Short>()\n for (e in this) {\n val key = selector(e)\n if
(set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the
given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting
list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> IntArray.distinctBy(selector: (Int) -> K): List<Int> {\n val set = HashSet<K>()\n val list = ArrayList<Int>()\n
for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n *
Returns a list containing only elements from
the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the
resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {\n val set = HashSet<K>()\n val list =
ArrayList<Long>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> FloatArray.distinctBy(selector: (Float) -> K): List<Float> {\n val set = HashSet<K>()\n
val list = ArrayList<Float>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having
distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same
order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {\n val set = HashSet<K>()\n val list =
ArrayList<Double>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they

```

were

```
in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n * \npublic inline fun <K> BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {\n val set =\n HashSet<K>()\n val list = ArrayList<Boolean>()\n for (e in this) {\n val key = selector(e)\n if\n (set.add(key))\n list.add(e)\n }\n return list\n}\n\n * Returns a list containing only elements from the\n * given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting\n * list are in the same order as they were in the source array.\n * \n * @sample\n * samples.collections.Collections.Transformations.distinctAndDistinctBy\n * \npublic inline fun <K>\n * CharArray.distinctBy(selector: (Char) -> K): List<Char> {\n * val set = HashSet<K>()\n * val list =\n * ArrayList<Char>()\n * for (e in this) {\n * val key = selector(e)\n * if (set.add(key))\n * list.add(e)\n * }\n * return\n * list\n * }\n\n * Returns a set containing all elements that are contained by both this array and the specified\n * collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set\n * containing all elements that are contained at least in one of these collections use [union].\n * \npublic infix fun <T>\n * Array<out T>.intersect(other: Iterable<T>): Set<T> {\n * val set = this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all elements that are contained by both this array and the specified\n * collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set\n * containing all elements that are contained at least in one of these collections use [union].\n * \npublic infix fun\n * ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {\n * val set = this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing\n * all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the\n * element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in\n * one of these collections use [union].\n * \npublic infix fun ShortArray.intersect(other: Iterable<Short>): Set<Short>\n * {\n * val set = this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all\n * elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the\n * element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in\n * one of these collections use [union].\n * \npublic infix fun IntArray.intersect(other: Iterable<Int>): Set<Int> {\n * val set = this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all\n * elements that are contained by both this\n * array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original\n * array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use\n * [union].\n * \npublic infix fun LongArray.intersect(other: Iterable<Long>): Set<Long> {\n * val set =\n * this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all elements that are\n * contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration\n * order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these\n * collections use [union].\n * \npublic infix fun FloatArray.intersect(other: Iterable<Float>): Set<Float> {\n * val set =\n * this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all elements that\n * are contained by both this array and the specified collection.\n * \n * \n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all\n * elements that are contained at least in one of these collections use [union].\n * \npublic infix fun\n * DoubleArray.intersect(other: Iterable<Double>): Set<Double> {\n * val set = this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all elements that are contained by both this\n * array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original\n * array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use\n * [union].\n * \npublic infix fun BooleanArray.intersect(other: Iterable<Boolean>): Set<Boolean> {\n * val set =\n * this.toMutableSet()\n * set.retainAll(other)\n * return set\n * }\n\n * Returns a set containing all elements that are\n * contained by both this array and the specified collection.\n * \n * The returned set preserves
```

the element iteration order of the original array.

To get a set containing all elements that are contained at least in one of these collections use [union].

```
public infix fun CharArray.intersect(other: Iterable<Char>): Set<Char> {
 val set = this.toMutableSet()
 set.retainAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun <T> Array<out T>.subtract(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun ByteArray.subtract(other: Iterable<Byte>): Set<Byte> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun IntArray.subtract(other: Iterable<Int>): Set<Int> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun LongArray.subtract(other: Iterable<Long>): Set<Long> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun FloatArray.subtract(other: Iterable<Float>): Set<Float> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun DoubleArray.subtract(other: Iterable<Double>): Set<Double> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
public infix fun BooleanArray.subtract(other: Iterable<Boolean>): Set<Boolean> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a set containing all elements that are contained by this array and not contained by the specified collection. The returned set preserves the element iteration order of the original array.

```
CharArray.subtract(other: Iterable<Char>): Set<Char> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original array.

```
<T> Array<out T>.toMutableSet(): MutableSet<T> {
 return toCollection(LinkedHashSet<T>(mapCapacity(size)))
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original array.

```
ByteArray.toMutableSet(): MutableSet<Byte> {
 return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original array.

```
ShortArray.toMutableSet(): MutableSet<Short> {
 return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original array.

```
IntArray.toMutableSet(): MutableSet<Int> {
 return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}
```

Returns a new [MutableSet] containing all distinct elements from the given array. The returned set preserves the element iteration order of the original array.

```
LongArray.toMutableSet(): MutableSet<Long> {
 return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}
```

Returns a new [MutableSet] containing all

distinct elements from the given array.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \*/\npublic fun FloatArray.toMutableSet(): MutableSet<Float> {\n return toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n}\n\n/\*\*\n \* Returns a new [MutableSet] containing all distinct elements from the given array.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \*/\npublic fun DoubleArray.toMutableSet(): MutableSet<Double> {\n return toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n}\n\n/\*\*\n \* Returns a new [MutableSet] containing all distinct elements from the given array.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \*/\npublic fun BooleanArray.toMutableSet(): MutableSet<Boolean> {\n return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n}\n\n/\*\*\n \* Returns a new [MutableSet] containing all distinct elements from the given array.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \*/\npublic fun CharArray.toMutableSet(): MutableSet<Char> {\n return toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128)))\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun IntArray.union(other: Iterable<Int>): Set<Int> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun LongArray.union(other: Iterable<Long>): Set<Long> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements from both collections.\n \* \n \* The returned set preserves the element iteration order of the original array.\n \* Those elements of the [other] collection that are unique are iterated in the end\n \* in the order of the [other] collection.\n \* \n \* To get a set containing all elements that are contained in both collections use [intersect].\n \*/\npublic infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/\*\*\n \* Returns a set containing all distinct elements

from both collections. The returned set preserves the element iteration order of the original array. Those elements of the [other] collection that are unique are iterated in the end in the order of the [other] collection. To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
 val set = this.toMutableSet()
 set.addAll(other)
 return set
}

```

Returns a set containing all distinct elements from both collections. The returned set preserves the element iteration order of the original array. Those elements of the [other] collection that are unique are iterated in the end in the order of the [other] collection. To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun CharArray.union(other: Iterable<Char>): Set<Char> {
 val set = this.toMutableSet()
 set.addAll(other)
 return set
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```

Returns `true` if all elements match the given [predicate]. Note that if the array contains no elements, the function returns `true` because there are no elements in it that `do not` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample

```

samples.collections.Collections.Aggregates.all
public inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
}

```





```

@sample samples.collections.Collections.Aggregates.anyWithPredicate\n *^\\npublic inline fun
DoubleArray.any(predicate: (Double) -> Boolean): Boolean {\\n for (element in this) if (predicate(element)) return
true\\n return false\\n}\\n\\n/**\\n * Returns `true` if at least one element matches the given [predicate].\\n * \\n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\\n
*^\\npublic inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {\\n for (element in this) if
(predicate(element)) return true\\n return false\\n}\\n\\n/**\\n * Returns `true` if at least one element matches the
given [predicate].\\n * \\n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\\n *^\\npublic
inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {\\n for (element in this) if (predicate(element))
return true\\n return false\\n}\\n\\n/**\\n * Returns the number of elements in this array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic inline fun <T> Array<out T>.count(): Int {\\n return size\\n}\\n\\n/**\\n *
Returns the number of elements in this array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic inline fun
ByteArray.count(): Int {\\n return size\\n}\\n\\n/**\\n * Returns the number of elements in this array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic inline fun ShortArray.count(): Int {\\n return size\\n}\\n\\n/**\\n
* Returns the number of elements in this array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic inline fun
IntArray.count(): Int {\\n return size\\n}\\n\\n/**\\n * Returns the number of elements in this array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic inline fun LongArray.count(): Int {\\n return size\\n}\\n\\n/**\\n * Returns
the number of elements in this array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic inline fun FloatArray.count(): Int {\\n
return size\\n}\\n\\n/**\\n * Returns the number of elements in this array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic
inline fun DoubleArray.count(): Int {\\n return size\\n}\\n\\n/**\\n * Returns the number of elements in this array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic inline fun BooleanArray.count(): Int {\\n return size\\n}\\n\\n/**\\n * Returns
the number of elements in this array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic inline fun CharArray.count(): Int {\\n
return size\\n}\\n\\n/**\\n * Returns the number of elements matching the given [predicate].\\n
*^\\npublic inline fun <T> Array<out T>.count(predicate: (T) -> Boolean): Int {\\n var count = 0\\n for (element in
this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the number of elements matching the
given [predicate].\\n *^\\npublic inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {\\n var count = 0\\n
for (element in this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the number of elements
matching the given [predicate].\\n *^\\npublic inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {\\n var
count = 0\\n for (element in this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the
number of elements matching the given [predicate].\\n *^\\npublic inline fun IntArray.count(predicate: (Int) ->
Boolean): Int {\\n var count = 0\\n for (element in this) if (predicate(element)) ++count\\n return
count\\n}\\n\\n/**\\n * Returns the number of elements matching the given
[predicate].\\n *^\\npublic inline fun LongArray.count(predicate: (Long) -> Boolean): Int {\\n var count = 0\\n for
(element in this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the number of elements
matching the given [predicate].\\n *^\\npublic inline fun FloatArray.count(predicate: (Float) -> Boolean): Int {\\n var
count = 0\\n for (element in this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the
number of elements matching the given [predicate].\\n *^\\npublic inline fun DoubleArray.count(predicate: (Double) -
> Boolean): Int {\\n var count = 0\\n for (element in this) if (predicate(element)) ++count\\n return
count\\n}\\n\\n/**\\n * Returns the number of elements matching the given [predicate].\\n *^\\npublic inline fun
BooleanArray.count(predicate: (Boolean) -> Boolean): Int {\\n var count = 0\\n for (element in this) if
(predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Returns the number of elements matching
the given [predicate].\\n *^\\npublic inline fun CharArray.count(predicate: (Char) -> Boolean): Int {\\n var count =
0\\n for (element in this) if (predicate(element)) ++count\\n return count\\n}\\n\\n/**\\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\\n * to current accumulator value and each element.\\n *
\\n * Returns the specified [initial] value if the array is empty.\\n * \\n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\\n *^\\npublic inline fun <T, R>
Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {\\n var accumulator = initial\\n for (element in this)
accumulator = operation(accumulator, element)\\n return accumulator\\n}\\n\\n/**\\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\\n * to current accumulator value and each element.\\n *

```

`\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left`



Char) -> R): R { \n var index = 0 \n var accumulator = initial \n for (element in this) accumulator = operation(index++, accumulator, element) \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <T, R> Array<out T>.foldRight(initial: R, operation: (T, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right to left \n \* to each element and current accumulator value. \n \* \n \* Returns the specified [initial] value if the array is empty. \n \* \n \* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. \n \* \n \n public inline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R { \n var index = lastIndex \n var accumulator = initial \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n \* Accumulates value starting with [initial] value and applying [operation] from right

to left to each element and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(get(index--), accumulator)
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation]

```
public inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation]

function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (index: Int, Double, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty.

@param [operation] function

that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
public inline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Performs the given [action] on each element.

```
public inline fun <T> Array<out T>.forEach(action: (T) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun ShortArray.forEach(action: (Short) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun IntArray.forEach(action: (Int) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun LongArray.forEach(action: (Long) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun FloatArray.forEach(action: (Float) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun BooleanArray.forEach(action: (Boolean) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element.

```
public inline fun CharArray.forEach(action: (Char) -> Unit): Unit {
 for (element in this) action(element)
}
```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```
public inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
 var index = 0
 for (item in this) action(index++, item)
}
```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```
public inline fun ByteArray.forEachIndexed(action: (index: Int, Byte) -> Unit): Unit {
 var index = 0
 for (item in this) action(index++, item)
}
```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```
public inline fun ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit): Unit {
 var index = 0
 for (item in this) action(index++, item)
}
```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```
public inline fun IntArray.forEachIndexed(action: (index: Int, Int) -> Unit): Unit {
 var index = 0
 for (item in this) action(index++, item)
}
```

Performs the given [action] on each element, providing sequential index with

the element.\n \* @param [action] function that takes the index of an element and the element itself\n \* and performs the action on the element.\n \*/\npublic inline fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/\*\*\n \* Performs the given [action] on each element, providing sequential index with the element.\n \* @param [action] function that takes the index of an element and the element itself\n \* and performs the action on the element.\n \*/\npublic inline fun FloatArray.forEachIndexed(action: (index: Int, Float) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/\*\*\n \* Performs the given [action] on each element, providing sequential index with the element.\n \* @param [action] function that takes the index of an element and the element itself\n \* and performs the action on the element.\n \*/\npublic inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/\*\*\n \* Performs the given [action] on each element, providing sequential index with the element.\n \* @param [action] function that takes the index of an element and the element itself\n \* and performs the action on the element.\n \*/\npublic inline fun BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/\*\*\n \* Performs the given [action] on each element, providing sequential index with the element.\n \* @param [action] function that takes the index of an element and the element itself\n \* and performs the action on the element.\n \*/\npublic inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* If any of elements is `NaN` returns `NaN`.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun Array<out Double>.max(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* If any of elements is `NaN` returns `NaN`.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun Array<out Float>.max(): Float {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun <T : Comparable<T>> Array<out T>.max(): T {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun ByteArray.max(): Byte {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun ShortArray.max(): Short {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")\npublic fun IntArray.max(): Int {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/\*\*\n \* Returns the largest element.\n \* \n \* @throws NoSuchElementException if the array is empty.\n \*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING\_OVERLOADS")

```

DS\)\npublic fun LongArray.max(): Long {\n if (isEmpty()) throw NoSuchElementException()\n var max =\n this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA\nDS\)\npublic fun FloatArray.max(): Float {\n if (isEmpty()) throw NoSuchElementException()\n var max =\n this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest\n * element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the\n * array is empty.\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA\nDS\)\npublic fun DoubleArray.max(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var max =\n this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA\nDS\)\npublic fun CharArray.max(): Char {\n if (isEmpty()) throw NoSuchElementException()\n var max =\n this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws\n * NoSuchElementException\n * if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL\nOADS\)\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T {\n if\n (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if\n (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =\n this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.maxBy\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL\nOADS\)\npublic\n inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte {\n if (isEmpty()) throw\n NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)\n return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v\n = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return\n maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.maxBy\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL\nOADS\)\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short {\n if\n (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex\n = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in\n 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value\n * of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.maxBy\n */\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL\nOADS\)\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int {\n if (isEmpty())\n throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==

```



```

0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * @throws NoSuchElementException if the array is empty.\n */\n\n@sample samples.collections.Collections.Aggregates.maxBy\n\n*\/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * @throws NoSuchElementException if the array is empty.\n */\n\n@sample samples.collections.Collections.Aggregates.maxBy\n\n*\/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * @throws NoSuchElementException if the array is empty.\n */\n\n@sample samples.collections.Collections.Aggregates.maxBy\n\n*\/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * @throws NoSuchElementException if the array is empty.\n */\n\n@sample samples.collections.Collections.Aggregates.maxBy\n\n*\/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n\n*\/\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxByOrNull(selector: (T) -> R): T? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex

```

```

== 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) -> R): Byte? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) -> R): Short? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R): Int? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.maxByOrNull\n *\n@SinceKotlin(\n"1.4\n")\npublic inline fun <R : Comparable<R>> CharArray.maxByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxV = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxV < v) {\n maxElem = e\n maxV = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n\n*\n@SinceKotlin(\n"1.4\n")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxV = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxV = maxOf(maxV, v)\n }\n return maxV\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */\n}\n

```

`maxValue` Returns the largest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws `NoSuchElementException` if the array is empty.

```

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun DoubleArray.maxOf(selector: (Double) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n**n * Returns the largest value among all values produced by [selector] function applied to
each element in the array.n * n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * n * @throws NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic
inline fun BooleanArray.maxOf(selector: (Boolean) -> Double): Double {n if (isEmpty()) throw
NoSuchElementException()n var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v =
selector(this[i])n maxValue = maxOf(maxValue, v)n }n return maxValue}n/n/n**n * Returns the
largest value among all values produced by [selector] function applied to each element in the array.n * n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n * n * @throws
NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun CharArray.maxOf(selector: (Char) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n
for (i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n**n * Returns the largest value among all values produced by [selector] function applied to
each element in the array.n * n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * n * @throws NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <T> Array<out T>.maxOf(selector: (T) ->
Float): Float {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for (i
in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n**n * Returns the largest value among all values produced by [selector] function applied to
each element in the array.n
* n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n * n * @throws
NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun ByteArray.maxOf(selector: (Byte) -> Float):
Float {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for (i in
1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n**n * Returns the largest value among all values produced by [selector] function applied to
each element in the array.n * n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * n * @throws NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic
inline fun ShortArray.maxOf(selector: (Short) -> Float): Float {n if (isEmpty()) throw
NoSuchElementException()n var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v =
selector(this[i])n maxValue = maxOf(maxValue, v)n }n return maxValue}n/n/n**n * Returns the
largest value among all values produced by [selector] function applied to each element in the array.n * n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n * n * @throws

```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Float):
Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in
1..lastIndex)
```

```
{\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n *
```

Returns the largest value among all values produced by [selector] function\n \* applied to each element in the  
array.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n \* \n \*

@throws NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->
```

```
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i
in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
```

```
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values
```

produced by [selector] function is `NaN`, the returned result is `NaN`.\n \* \n \* @throws NoSuchElementException  
if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) ->
```

```
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i
in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
```

```
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
```

```
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun DoubleArray.maxOf(selector: (Double) -> Float): Float {\n if (isEmpty()) throw
```

```
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n *
```

Returns the  
largest value among all values produced by [selector] function\n \* applied to each element in the array.\n \* \n \* If

any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n \* \n \* @throws

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) ->
```

```
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i
in 1..lastIndex) {\n val v
```

```
= selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the
largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If
```

any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n \* \n \* @throws

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) -> Float):
```

```
Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i
in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
```

```
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * @throws NoSuchElementException
```

if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out
T>.maxOf(selector: (T) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ByteArray.maxOf(selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n
 var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue <
v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ShortArray.maxOf(selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element
in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.maxOf(selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.maxOf(selector: (Long) -> R):
R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n
return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.maxOf(selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOf(selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws

```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R
```

```
: Comparable<R>> BooleanArray.maxOf(selector: (Boolean) -> R): R {\n if (isEmpty()) throw\n NoSuchElementException()\n var max\n Value = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (max\n Value < v) {\n max\n Value = v\n }\n }\n return max\n Value\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
CharArray.maxOf(selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var\n max\n Value = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (max\n Value < v) {\n max\n Value = v\n }\n }\n return max\n Value\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the\n * returned result is `NaN`.\n */
```

```
* Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the\n * returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
```

```
(T) -> Double): Double? {\n if (isEmpty()) return null\n var max\n Value = selector(this[0])\n for (i in\n 1..lastIndex) {\n val v = selector(this[i])\n max\n Value = maxOf(max\n Value, v)\n }\n return\n max\n Value\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Double): Double? {\n if (isEmpty()) return null\n var\n max\n Value = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n max\n Value =\n maxOf(max\n Value, v)\n }\n return max\n Value\n}\n\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of\n * values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -
```

```
> Double): Double? {\n if (isEmpty()) return null\n var max\n Value = selector(this[0])\n for (i\n in 1..lastIndex) {\n val v = selector(this[i])\n max\n Value = maxOf(max\n Value, v)\n }\n return\n max\n Value\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to\n * each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) ->
```

```
Double): Double? {\n if (isEmpty()) return null\n var max\n Value = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n max\n Value = maxOf(max\n Value, v)\n }\n return max\n Value\n}\n\n/**\n * Returns\n * the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`\n * if there are no elements.\n * \n * \n * \n */
```

```
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOfOrNull(selector: (Long) -
```

```
> Double): Double? {\n if (isEmpty()) return null\n var max\n Value = selector(this[0])\n for (i in 1..lastIndex)\n {\n val v = selector(this[i])\n max\n Value = maxOf(max\n Value, v)\n }\n return max\n Value\n}\n\n/**\n * \n * \n * \n */
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector:\n(Float) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return\n maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOrNull(selector:\n(Double) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return\n maxValue\n}\n/**\n * Returns the largest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector:\n(Boolean) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return\n maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\n inline fun CharArray.maxOrNull(selector: (Char) -> Double): Double? {\n if (isEmpty()) return null\n var\n maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =\n maxOf(maxValue, v)\n }\n return maxValue\n}\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any of\n * values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOrNull(selector:\n(T) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)\n {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n/**\n * Returns the largest value among all\n * values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * If any\n * of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOrNull(selector: (Byte) ->\nFloat): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n/**\n * Returns\n * the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`\n * if there are no elements.\n * If any of values produced by [selector] function\n * is\n * `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOrNull(selector: (Short) ->\n> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
```



```

val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOrNull(selector: (Int) ->
Float): Float? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue
= maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOrNull(selector: (Long) -
> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in
the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector: (Float) -
> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.maxOrNull(selector: (Double) -> Float): Float? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector:
(Boolean) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) ->
Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

```

```
inline fun <T, R : Comparable<R>> Array<out T>.maxOrNull(selector: (T) -> R): R? {\n if (isEmpty()) return\n null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if\n (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value\n among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no\n elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
ByteArray.maxOrNull(selector: (Byte) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n
```

```
* Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
ShortArray.maxOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
IntArray.maxOrNull(selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
LongArray.maxOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector]
```

```
function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
FloatArray.maxOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
DoubleArray.maxOrNull(selector: (Double) -> R): R? {\n if\n (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array\n or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```
BooleanArray.maxOrNull(selector: (Boolean) -> R): R? {\n if (isEmpty()) return null\n var maxValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
```

```

maxValue = v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWith(comparator: Comparator<in R>, selector: (T)
-> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue
= v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.maxOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue
= v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n}\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.maxOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.maxOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function
applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.maxOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n if
(isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n
return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all
values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to
each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if

```

```

(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nLongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nFloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nDoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nBooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n if (isEmpty()) return null\n var\n maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nCharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there

```

are no elements. If any of elements is `NaN` returns `NaN`.

```

@SinceKotlin("1.4")
public fun Array<out Double>.maxOrNull(): Double? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 max = maxOf(max, e)
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun Array<out Float>.maxOrNull(): Float? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 max = maxOf(max, e)
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun <T : Comparable<T>> Array<out T>.maxOrNull(): T? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun ByteArray.maxOrNull(): Byte? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun ShortArray.maxOrNull(): Short? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun IntArray.maxOrNull(): Int? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun LongArray.maxOrNull(): Long? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

If any of elements is `NaN` returns `NaN`.

```

@SinceKotlin("1.4")
public fun FloatArray.maxOrNull(): Float? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 max = maxOf(max, e)
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

If any of elements is `NaN` returns `NaN`.

```

@SinceKotlin("1.4")
public fun DoubleArray.maxOrNull(): Double? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 max = maxOf(max, e)
 }
 return max
}

```

Returns the largest element or `null` if there are no elements.

```

@SinceKotlin("1.4")
public fun CharArray.maxOrNull(): Char? {
 if (isEmpty()) return null
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

```

Returns the first element having the largest value according to the provided [comparator].

```

@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("maxWithOrThrow")
@Suppress("CONFLICTING_OVERLOADS")
public fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T {
 if (isEmpty()) throw NoSuchElementException()
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(max, e) < 0) max = e
 }
 return max
}

```

Returns the first element having the largest value according to the provided [comparator].

```

@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("maxWithOrThrow")
@Suppress("CONFLICTING_OVERLOADS")
public fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte {
 if (isEmpty()) throw NoSuchElementException()
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(max, e) < 0) max = e
 }
 return max
}

```

Returns the first element having the largest value according to the provided [comparator].

```

@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("maxWithOrThrow")
@Suppress("CONFLICTING_OVERLOADS")
public fun ShortArray.maxWith(comparator: Comparator<in Short>): Short {
 if (isEmpty()) throw NoSuchElementException()
 var max = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(max, e) < 0) max = e
 }
 return max
}

```

Returns the

```

first element having the largest value according to the provided [comparator].\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun IntArray.maxWith(comparator: Comparator<in Int>): Int {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun LongArray.maxWith(comparator: Comparator<in Long>): Long {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n
if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element
having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun FloatArray.maxWith(comparator: Comparator<in Float>): Float {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun DoubleArray.maxWith(comparator: Comparator<in Double>): Double {\n if (isEmpty())
throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun BooleanArray.maxWith(comparator: Comparator<in Boolean>): Boolean {\n if
(isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e =
this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first
element having the largest value according to the provided [comparator].\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic
fun CharArray.maxWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun <T> Array<out T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun ByteArray.maxWithOrNull(comparator: Comparator<in Byte>): Byte? {\n
if (isEmpty())
return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max,
e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun
ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {\n if (isEmpty()) return null\n var max

```

```

= this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator]\n or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.maxWithOrNull(comparator:\n Comparator<in Int>): Int? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val\n e = this[i]\n if (comparator.compare(max, e)\n < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the\n provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n if (isEmpty()) return null\n var max\n = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator]\n or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n FloatArray.maxWithOrNull(comparator:\n Comparator<in Float>): Float? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val\n e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the\n first element having the largest value according to the provided [comparator]\n or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {\n if (isEmpty()) return null\n var max\n = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max =\n e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided\n [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n if (isEmpty()) return null\n var max\n = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max =\n e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided\n [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun\n CharArray.maxWithOrNull(comparator:\n Comparator<in Char>): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val\n e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the\n smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\n DS")\npublic fun Array<out Double>.min(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var\n min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return\n min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\n DS")\npublic fun Array<out Float>.min():\n Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val\n e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws\n NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\n DS")\npublic fun <T : Comparable<T>> Array<out T>.min(): T {\n if (isEmpty()) throw\n NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e)\n min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws\n NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\n DS")\npublic fun ByteArray.min(): Byte {\n if (isEmpty()) throw NoSuchElementException()\n var min =\n this[0]\n for (i in 1..lastIndex) {\n val\n e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws\n NoSuchElementException if the array is empty.\n */

```



```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ShortArray.min(): Short {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun IntArray.min(): Int {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException\n * if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun LongArray.min(): Long {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun FloatArray.min(): Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun DoubleArray.min(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharArray.min(): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic\n inline fun <T, R : Comparable<R>> Array<out T>.minBy(selector: (T) -> R): T {\n if (isEmpty()) throw\n NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)\n return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v =\n selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return\n minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> ByteArray.minBy(selector: (Byte) -> R): Byte {\n if (isEmpty())\n throw\n NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)\n return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v =\n selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return\n minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")

```

```

ADS`)npublic inline fun <R : Comparable<R>> ShortArray.minBy(selector: (Short) -> R): Short {n if
(isEmpty()) throw NoSuchElementException()n var minElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return minElemn var minValue = selector(minElem)n for (i in 1..lastIndex)
{n val e = this[i]n val v = selector(e)n if (minValue > v) {n minElem = e\n minValue
= v\n }}n }n return minElem\n}\n\n/**n * Returns the first element yielding the smallest value of the given
function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS`)npublic inline fun <R : Comparable<R>> IntArray.minBy(selector: (Int) -> R): Int {n if (isEmpty())
throw NoSuchElementException()n var minElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex ==
0) return minElemn var minValue = selector(minElem)n for (i in 1..lastIndex) {n val e = this[i]n val
v = selector(e)n if (minValue > v) {n minElem = e\n minValue = v\n }}n }n return
minElem\n}\n\n/**n * Returns
the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if
the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS`)npublic inline fun <R : Comparable<R>> LongArray.minBy(selector: (Long) -> R): Long {n if
(isEmpty()) throw NoSuchElementException()n var minElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return minElemn var minValue = selector(minElem)n for (i in 1..lastIndex) {n val e =
this[i]n val v = selector(e)n if (minValue > v) {n minElem = e\n minValue = v\n }}n }n return
minElem\n}\n\n/**n * Returns the first element yielding the smallest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS`)npublic inline fun <R : Comparable<R>> FloatArray.minBy(selector: (Float) -> R): Float {n if
(isEmpty()) throw NoSuchElementException()n var minElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return minElemn var minValue = selector(minElem)n for (i in 1..lastIndex) {n val e =
this[i]n val v = selector(e)n if (minValue > v) {n minElem = e\n minValue = v\n }}n }n return
minElem\n}\n\n/**n * Returns the first element yielding the smallest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS`)npublic inline fun <R : Comparable<R>> DoubleArray.minBy(selector: (Double) ->
R): Double {n if (isEmpty()) throw NoSuchElementException()n var minElem = this[0]n val lastIndex =
this.lastIndexn if (lastIndex == 0) return minElemn var minValue = selector(minElem)n for (i in
1..lastIndex) {n val e = this[i]n val v = selector(e)n if (minValue > v) {n minElem = e\n
minValue = v\n }}n }n return minElem\n}\n\n/**n * Returns the first element yielding the smallest value
of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS`)npublic inline fun <R : Comparable<R>> BooleanArray.minBy(selector: (Boolean) -> R): Boolean {n if
(isEmpty()) throw NoSuchElementException()n var minElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return minElemn var minValue
= selector(minElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (minValue >
v) {n minElem = e\n minValue = v\n }}n }n return minElem\n}\n\n/**n * Returns the first
element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n
*/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO

```

```

ADS")\npublic inline fun <R : Comparable<R>> CharArray.minBy(selector: (Char) -> R): Char {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function
or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Array<out T>.minByOrNull(selector: (T) ->
R): T? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex
== 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n
val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there
are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ByteArray.minByOrNull(selector:
(Byte) -> R): Byte? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n
if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ShortArray.minByOrNull(selector: (Short) ->
R): Short? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R): Int? {\n
if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)
return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v =
selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> LongArray.minByOrNull(selector: (Long) ->
R): Long? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> FloatArray.minByOrNull(selector: (Float) ->
R): Float? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.minByOrNull(selector: (Double) -> R):
Double? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first

```

```

element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.minByOrNull(selector:
(Boolean) -> R): Boolean? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex =
this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in
1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n
minValue = v\n }\n }\n return minElem\n}\n\n**\n * Returns the first element yielding the smallest value
of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n *\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var
minElem = this[0]\n val lastIndex
= this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in
1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n
minValue = v\n }\n }\n return minElem\n}\n\n**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.minOf(selector: (Int) -> Double): Double {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) ->

```

```

Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun FloatArray.minOf(selector: (Float) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun BooleanArray.minOf(selector: (Boolean) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.minOf(selector: (Char) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ByteArray.minOf(selector: (Byte) -> Float): Float {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

```

\* Returns the smallest value among all values produced by [selector] function applied to each element in the array. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the array is empty.

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector:
(Short) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Float):
Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) ->
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Float):
Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector:
(Double) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) ->
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Float):
Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return

```

```

minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\ninline fun <T, R : Comparable<R>> Array<out T>.minOf(selector: (T) -> R): R {\n if (isEmpty()) throw\n NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nByteArray.minOf(selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue\n = selector(this[0])\n for (i in 1..lastIndex) {\n val\n v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nShortArray.minOf(selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var\n minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n \n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nIntArray.minOf(selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue =\n selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n \n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the\n * array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nLongArray.minOf(selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var\n minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n \n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nFloatArray.minOf(selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var\n minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n \n minValue = v\n }\n }\n return minValue}\n\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array.\n * \n * @throws\n * NoSuchElementException if the array is empty.\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nDoubleArray.minOf(selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var\n minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n \n minValue = v\n }\n }\n return minValue}\n\n}\n
```

```

 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.minOf(selector: (Boolean) -> R): R {\n if (isEmpty())
throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.minOf(selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector]
function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:
(T) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.minOfOrNull(selector: (Byte) -> Double): Double? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -
> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) ->
Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -

```



> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null`\n if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) -> Double): Double? {\n if (isEmpty()) return null\n

var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the

array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

inline fun CharArray.minOfOrNull(selector: (Char) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

\*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue}\n\n/n/\*\*\n \* Returns the smallest value among all values produced by [selector] function\n \* applied to each element in the array or `null` if there are no elements.\n \* \n \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.minOrNull(selector: (Short) -> Float): Float? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOrNull(selector: (Int) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOrNull(selector: (Long) -
> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n * \n * If any
of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOrNull(selector: (Float) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOrNull(selector:
(Double) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOrNull(selector:
(Boolean) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOrNull(selector: (Char) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
 Array<out T>.minOfOrNull(selector: (T) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
 by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 ByteArray.minOfOrNull(selector: (Byte) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
 by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 ShortArray.minOfOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
 by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 IntArray.minOfOrNull(selector: (Int) -> R): R? {\n
 if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
 selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
 or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 LongArray.minOfOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
 by [selector] function\n * applied to each element in the
 array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 FloatArray.minOfOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
 by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 DoubleArray.minOfOrNull(selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var minValue =
 selector(this[0])\n
 for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n
 }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
 applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
 BooleanArray.minOfOrNull(selector: (Boolean) -> R): R? {\n if (isEmpty()) return null\n var minValue =

```

```

selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharArray.minOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n
 var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.minOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue,
v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWith(comparator:
Comparator<in R>, selector:
(Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n *
\n * @throws NoSuchElementException if the array is empty.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.minOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n
\n * @throws NoSuchElementException if the array is empty.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n
\n * @throws NoSuchElementException if the array is empty.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.minOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n if
(isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
 val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n
 return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all
values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n
 var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue,
 v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array or `null` if there are no elements.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null`
if there are no elements.\n
\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n if (isEmpty())

```

```

return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun Array<out Double>.minOrNull(): Double? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n
```

```

min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no
elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun Array<out
Float>.minOrNull(): Float? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val
e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if
there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Array<out T>.minOrNull():
T? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min
> e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun
ByteArray.minOrNull(): Byte? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val
e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if
there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun ShortArray.minOrNull(): Short? {\n if (isEmpty())
return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n if (isEmpty()) return null\n var min =
this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no
elements.\n */\n@SinceKotlin("1.4")\npublic fun LongArray.minOrNull(): Long? {\n if (isEmpty()) return
null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return
min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is
`NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.minOrNull(): Float? {\n if (isEmpty())
return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is
`NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull(): Double? {\n if
(isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min,
e)\n }\n return min\n}\n\n/**\n * Returns
the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun
CharArray.minOrNull(): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val
e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the
array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun ShortArray.minWith(comparator: Comparator<in Short>): Short {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having
the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array
is empty.\n */

```

```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun IntArray.minWith(comparator: Comparator<in Int>): Int {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun LongArray.minWith(comparator: Comparator<in Long>): Long {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun FloatArray.minWith(comparator: Comparator<in Float>): Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun DoubleArray.minWith(comparator: Comparator<in Double>): Double {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharArray.minWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */
*\n@SinceKotlin("1.4")\npublic fun <T> Array<out T>.minWithOrNull(comparator: Comparator<in T>): T? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */
*\n@SinceKotlin("1.4")\npublic fun ByteArray.minWithOrNull(comparator: Comparator<in Byte>): Byte? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */
*\n@SinceKotlin("1.4")\npublic fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */

```



```

or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun IntArray.minWithOrNull(comparator:
Comparator<in Int>): Int? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val
e = this[i]\n if (comparator.compare(min, e) > 0) min
= e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator] or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun
LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {\n if (isEmpty()) return null\n var min
= this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n
return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator]
or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun FloatArray.minWithOrNull(comparator:
Comparator<in Float>): Float? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first
element having the smallest value according to the provided [comparator]
or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun
DoubleArray.minWithOrNull(comparator: Comparator<in Double>): Double? {\n if (isEmpty()) return null\n
var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min =
e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator] or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun
BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n if (isEmpty()) return null\n
var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min =
e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator] or `null` if there are no elements.\n * \n @SinceKotlin("1.4")\npublic fun
CharArray.minWithOrNull(comparator: Comparator<in
Char>): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if the array has no
elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n * \n public fun <T> Array<out
T>.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n * \n public fun ByteArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n * \n public fun ShortArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n * \n public fun IntArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
* \n public fun LongArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no
elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n * \n public fun FloatArray.none():
Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n * \n public fun DoubleArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n * \n public fun BooleanArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n * \n public fun CharArray.none(): Boolean {\n return
isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the
given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n * \n public
inline fun <T> Array<out T>.none(predicate: (T) -> Boolean): Boolean {\n for (element in this) if
(predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n * \n public inline fun
ByteArray.none(predicate: (Byte) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun ShortArray.none(predicate:
(Short) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return false\n return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*/\npublic inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {\n for (element in this) if
(predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
LongArray.none(predicate: (Long) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun FloatArray.none(predicate:
(Float) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return false\n return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
DoubleArray.none(predicate: (Double) -> Boolean): Boolean {\n for (element in this) if (predicate(element))
return false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (predicate(element))
return false\n return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun CharArray.none(predicate:
(Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return false\n return
true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.onEach(action: (T) ->
Unit): Array<out
T> {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each
element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.onEach(action: (Byte) -> Unit): ByteArray {\n return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEach(action:
(Short) -> Unit): ShortArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the
given [action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEach(action: (Int) -> Unit):
IntArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on
each element and returns
the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.onEach(action: (Long) -> Unit): LongArray {\n return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEach(action: (Float) ->
Unit): FloatArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEach(action: (Double) ->
Unit): DoubleArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEach(action:
(Boolean) -> Unit): BooleanArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n *
Performs the given [action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEach(action: (Char) ->
Unit): CharArray {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n
*/

```

```

@param [action] function that takes the index of an element and the element itself\n * and performs the action on
the element.\n *^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.onEachIndexed(action: (index: Int, T) -> Unit): Array<out T> {\n return apply { forEachIndexed(action)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the
array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n *
and performs the action on the element.\n *^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.onEachIndexed(action: (index: Int, Byte) -> Unit): ByteArray {\n return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEachIndexed(action:
(index: Int, Short) -> Unit): ShortArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the
given [action] on each element, providing sequential index with the element,\n * and returns the array itself
afterwards.\n * @param [action] function
that takes the index of an element and the element itself\n * and performs the action on the element.\n
*^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEachIndexed(action: (index:
Int, Int) -> Unit): IntArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action]
on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n *
@param [action] function that takes the index of an element and the element itself\n * and performs the action on
the element.\n *^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.onEachIndexed(action: (index: Int, Long) -> Unit): LongArray {\n return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element
itself\n * and performs the action on the element.\n *^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.onEachIndexed(action: (index: Int, Float) -> Unit): FloatArray {\n return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEachIndexed(action:
(index: Int, Double) -> Unit): DoubleArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs
the given [action] on each element, providing sequential index with the element,\n * and returns the array itself
afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the
element.\n *^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.onEachIndexed(action: (index: Int, Boolean) -> Unit): BooleanArray {\n return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*^n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEachIndexed(action:
(index: Int, Char) -> Unit): CharArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates
value starting with the first element and applying [operation] from left to right\n * to current accumulator value and
each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n *
please use [reduceOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an
element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n *^npublic inline fun <S, T : S> Array<out

```

```

T>.reduce(operation: (acc: S, T) -> S): S {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator: S = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduce

```

```

public inline fun ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduce

```

```

public inline fun ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduce

```

```

public inline fun IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduce

```

```

public inline fun LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduce

```

```

public inline fun FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 @sample

and calculates the next accumulator value.

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator: S = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(index, accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {
 if (isEmpty())
 throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = this[0]
 for (index in 1..lastIndex) {
 accumulator = operation(index, accumulator, this[index])
 }
 return accumulator
}

```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.reduce
*
public inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {
 if (isEmpty())
 throw

```

```

UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun LongArray.reduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): Long {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and

```

applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Throws an exception if this array is empty. If the array can be empty in an expected way,\n \* please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduce\n \* \n \* \n \* @public inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n \* \n \* \n \* @SinceKotlin("1.4")\n \* \n \* @public inline fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n \* \n \* \n \* @SinceKotlin("1.4")\n \* \n \* @public inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n \* \n \* \n \* @SinceKotlin("1.4")\n \* \n \* @public inline fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n \* \n \* \n \* @SinceKotlin("1.4")\n \* \n \* @public inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n \* \n \* \n \* @SinceKotlin("1.4")\n \* \n \* @public inline fun LongArray.reduceIndexedOrNull(operation: (index: Int, acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator}\n \* \n \* \n \* Accumulates value starting with the first element and applying [operation] from left to right\n \* to current accumulator value and each element with

its index in the original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n

```

*\n@SinceKotlin("1.4")\npublic inline fun FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator

```



```

value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {\n if (isEmpty())\n return null\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/>\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/>\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
LongArray.reduceOrNull(operation: (acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return null\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator
= operation(accumulator, this[index])\n }\n return accumulator}\n\n/>\n * Accumulates value starting with
the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n *
\n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value
and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/>\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each
element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return
null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator,
this[index])\n }\n return accumulator}\n\n/>\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(accumulator, this[index])\n }\n return accumulator}\n\n/>\n * Accumulates value starting with the
first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n

```

```

*^@SinceKotlin("1.4")^@WasExperimental(ExperimentalStdlibApi::class)^npublic inline fun
CharArray.reduceOrNull(operation: (acc: Char, Char) ->
Char): Char? {^n if (isEmpty())^n return null^n var accumulator = this[0]^n for (index in 1..lastIndex) {^n
accumulator = operation(accumulator, this[index])^n }^n return accumulator^n}^n/n/**^n * Accumulates value
starting with the last element and applying [operation] from right to left^n * to each element and current accumulator
value.^n * ^n * Throws an exception if this array is empty. If the array can be empty in an expected way,^n * please
use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.^n * ^n * @param [operation] function
that takes an element and current accumulator value,^n * and calculates the next accumulator value.^n * ^n *
@sample samples.collections.Collections.Aggregates.reduceRight^n *^npublic inline fun <S, T : S> Array<out
T>.reduceRight(operation: (T, acc: S) -> S): S {^n var index = lastIndex^n if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")^n var accumulator:
S = get(index--)^n while (index >= 0) {^n accumulator = operation(get(index--), accumulator)^n }^n return
accumulator^n}^n/n/**^n * Accumulates value starting with the last element and applying [operation] from right to
left^n * to each element and current accumulator value.^n * ^n * Throws an exception if this array is empty. If the
array can be empty in an expected way,^n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.^n * ^n * @param [operation] function that takes an element and current accumulator value,^n *
and calculates the next accumulator value.^n * ^n * @sample
samples.collections.Collections.Aggregates.reduceRight^n *^npublic inline fun ByteArray.reduceRight(operation:
(Byte, acc: Byte) -> Byte): Byte {^n var index = lastIndex^n if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")^n var accumulator = get(index--)^n while
(index >= 0) {^n accumulator = operation(get(index--),
accumulator)^n }^n return accumulator^n}^n/n/**^n * Accumulates value starting with the last element and
applying [operation] from right to left^n * to each element and current accumulator value.^n * ^n * Throws an
exception if this array is empty. If the array can be empty in an expected way,^n * please use [reduceRightOrNull]
instead. It returns `null` when its receiver is empty.^n * ^n * @param [operation] function that takes an element and
current accumulator value,^n * and calculates the next accumulator value.^n * ^n * @sample
samples.collections.Collections.Aggregates.reduceRight^n *^npublic inline fun ShortArray.reduceRight(operation:
(Short, acc: Short) -> Short): Short {^n var index = lastIndex^n if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")^n var accumulator = get(index--)^n while
(index >= 0) {^n accumulator = operation(get(index--), accumulator)^n }^n return accumulator^n}^n/n/**^n *
Accumulates
value starting with the last element and applying [operation] from right to left^n * to each element and current
accumulator value.^n * ^n * Throws an exception if this array is empty. If the array can be empty in an expected
way,^n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.^n * ^n * @param
[operation] function that takes an element and current accumulator value,^n * and calculates the next accumulator
value.^n * ^n * @sample samples.collections.Collections.Aggregates.reduceRight^n *^npublic inline fun
IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {^n var index = lastIndex^n if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")^n var accumulator = get(index--)^n while
(index >= 0) {^n accumulator = operation(get(index--), accumulator)^n }^n return accumulator^n}^n/n/**^n *
Accumulates value starting with the last element and applying [operation] from right to left^n
* to each element and current accumulator value.^n * ^n * Throws an exception if this array is empty. If the array
can be empty in an expected way,^n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is
empty.^n * ^n * @param [operation] function that takes an element and current accumulator value,^n * and calculates
the next accumulator value.^n * ^n * @sample samples.collections.Collections.Aggregates.reduceRight^n *^npublic
inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {^n var index = lastIndex^n if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")^n var accumulator =
get(index--)^n while (index >= 0) {^n accumulator = operation(get(index--), accumulator)^n }^n return
accumulator^n}^n/n/**^n * Accumulates value starting with the last element and applying [operation] from right to

```

left to each element and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRight
public inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(get(index--), accumulator)
 }
 return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRight
public inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(get(index--), accumulator)
 }
 return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRight
public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(get(index--), accumulator)
 }
 return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRight
public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(get(index--), accumulator)
 }
 return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

`@param [operation]` function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRightIndexed
public inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator: S = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use

[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

```

 * @param [operation]
 function that takes the
 index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator
 value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 *
 public inline fun
 ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator =
 get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
 }
 *
 Accumulates value starting with the last element and applying [operation]
 from right to left
 * to each element with its index in the original array and current accumulator value.
 *
 Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use
 [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 *
 * @param
 [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and
 calculates the next accumulator value.
 *
 * @sample samples.collections.Collections.Aggregates.reduceRight
 *
 public inline fun
 ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index),
 accumulator)
 --index
 }
 return accumulator
 }
 *
 Accumulates value starting with the last
 element and applying [operation] from right to left
 * to each element with its index in the original array and
 current accumulator value.
 *
 Throws an exception if this array is empty. If the array can be empty in an
 expected way,
 * please use [reduceRightIndexedOrNull] instead. It returns `null`
 when its receiver is empty.
 *
 * @param [operation] function that takes the index of an element, the element
 itself and current accumulator value,
 * and calculates the next accumulator value.
 *
 * @sample
 samples.collections.Collections.Aggregates.reduceRight
 *
 public inline fun
 IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {
 var index = lastIndex
 if (index
 < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return
 accumulator
 }
 *
 Accumulates value starting with the last element and applying [operation] from right to
 left
 * to each element with its index in the original array and current accumulator value.
 *
 Throws an
 exception if this array is empty. If the array can be empty in an expected way,
 * please use
 [reduceRightIndexedOrNull]
 instead. It returns `null` when its receiver is empty.
 *
 * @param [operation] function that takes the index of an
 element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 *
 * @sample
 samples.collections.Collections.Aggregates.reduceRight
 *
 public inline fun
 LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {
 var index =
 lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var
 accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index),
 accumulator)
 --index
 }
 return accumulator
 }
 *
 Accumulates value starting with the last
 element and applying [operation] from right to left
 * to each element with its index in the original array and
 current accumulator value.
 *
 Throws an exception if this array is empty. If the array can be empty in an
 expected way,
 *
 * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 *
 * @param
 [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and
 calculates the next accumulator value.
 *
 * @sample samples.collections.Collections.Aggregates.reduceRight
 *
 public inline fun
 FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index),
 accumulator)
 --index
 }
 return accumulator
 }
 *
 Accumulates value starting with the last
 element and applying [operation] from right to left
 * to each element with its index in the original array and

```

current accumulator value.  
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.  
 @sample  
 samples.collections.Collections.Aggregates.reduceRight

```

public inline fun DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.  
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.  
 @sample  
 samples.collections.Collections.Aggregates.reduceRight

```

public inline fun BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.  
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.  
 @sample  
 samples.collections.Collections.Aggregates.reduceRight

```

public inline fun CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.  
 Returns `null` if the array is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.  
 @sample  
 samples.collections.Collections.Aggregates.reduceRightOrNull

```

@SinceKotlin("1.4")
public inline fun <S, T : S> Array<out T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator: S = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.  
 Returns `null` if the array is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.  
 @sample  
 samples.collections.Collections.Aggregates.reduceRightOrNull

```

@SinceKotlin("1.4")
public inline fun ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.  
 Returns `null` if the array is empty.  
 @param [operation] function that takes the index of an element, the element itself and current accumulator value,

\* and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns `null` if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns `null` if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun LongArray.reduceRightIndexedOrNull(operation: (index: Int, Long, acc: Long) -> Long): Long? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns `null` if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index,
 get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns `null` if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double): Double? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns `null` if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

 @sample samples.collections.Collections.Aggregates.reduceRightOrNull
*/
@SinceKotlin("1.4")
public inline fun BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {
 var index = lastIndex
 if (index < 0) return null
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array



Returns `null` if the array is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```

samples.collections.Collections.Aggregates.reduceRightOrNull

*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {\n var index = lastIndex\n if
(index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator =
operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull

*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
inline fun DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {\n var index
= lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull

*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {\n var index =
lastIndex\n if (index < 0) return null\n
 var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--),
accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull

*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n var index = lastIndex\n if (index
< 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator =
operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningFold

*/\n@SinceKotlin("1.4")\npublic inline fun
<T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold

*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFold(initial: R,

```



operation: (acc: R, Byte) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n\*\*\n \* Returns a list containing successive accumulation

values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFold(initial: R, operation: (acc: R, Short) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive accumulation
```

values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFold(initial: R, operation: (acc: R, Int) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive
```

accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFold(initial: R, operation: (acc: R, Long) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive
```

accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.runningFold(initial: R, operation: (acc: R, Float) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive
```

accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

```
@sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.runningFold(initial: R,
operation: (acc: R, Double) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result =
ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n
accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n *
Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.runningFold(initial:
R, operation: (acc: R, Boolean) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result =
ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n
accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n *
Returns a
list containing successive accumulation values generated by applying [operation] from left to right\n * to each
element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.runningFold(initial: R,
operation: (acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator =
operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns
a list containing successive accumulation values generated by applying [operation] from left to right\n * to each
element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous
value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator
value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\npublic inline fun <T, R>
Array<out T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator
= initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function
```

```

function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {\n if (isEmpty())
return
 listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.runningFoldIndexed(initial:
R, operation: (index: Int, acc: R, Int) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result =
ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n
accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return
result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element,
current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
LongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself,
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =

```

initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/\*\*\n \* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should

not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
BooleanArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator
```

value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
```

```
CharArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
```

[operation] from left to right\n \* to each element and current accumulator value that starts with the first element of this array.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting list.\n \* \n \* @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n \* \n \* @sample

```
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
```

```
Array<out T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n if (isEmpty()) return emptyList()\n var accumulator: S = this[0]\n val result = ArrayList<S>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing
```

successive accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

```
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduce(operation:
```

```
(acc: Byte, Byte) -> Byte): List<Byte> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Byte>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator
```

value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* @sample

```
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.runningReduce(operation:
(acc: Short, Short) -> Short): List<Short> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Short>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator
```

```
value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.runningReduce(operation: (acc:
Int, Int) -> Int): List<Int> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result =
ArrayList<Int>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.runningReduce(operation:
(acc: Long, Long) -> Long): List<Long> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Long>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.runningReduce(operation:
(acc:
Float, Float) -> Float): List<Float> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val
result = ArrayList<Float>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.runningReduce(operation:
(acc: Double, Double) -> Double): List<Double> {\n if (isEmpty()) return emptyList()\n var accumulator =
this[0]\n val
result = ArrayList<Double>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.runningReduce(operation:
(acc: Boolean, Boolean) -> Boolean): List<Boolean> {\n if (isEmpty()) return emptyList()\n var accumulator =
this[0]\n val result = ArrayList<Boolean>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator
```

```

= operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a
list containing successive accumulation values generated by applying [operation] from left to right\n * to each
element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.runningReduce(operation:
(acc: Char, Char) -> Char): List<Char> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val
result = ArrayList<Char>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns
a list containing successive accumulation values generated by applying [operation] from left to right\n * to each
element, its index in the original array and current accumulator value that starts with the first element of this array.\n
*\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\npublic inline fun <S, T :
S> Array<out T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n if (isEmpty()) return
emptyList()\n var accumulator: S = this[0]\n val result = ArrayList<S>(size).apply { add(accumulator) }\n for
(index in 1 until size) {\n accumulator = operation(index, accumulator,
this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original
array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {\n if (isEmpty())
return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Byte>(size).apply { add(accumulator)
}\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n }\n result.add(accumulator)\n
}\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element,
current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {\n if
(isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Short>(size).apply {
add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n
 }\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element,
current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {\n if (isEmpty()) return
emptyList()\n var accumulator = this[0]\n val result = ArrayList<Int>(size).apply { add(accumulator) }\n for
(index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n }\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns

```

a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n @SinceKotlin("1.4")\n @kotlin.internal.InlineOnly\n public inline fun

LongArray.runningReduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): List<Long> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Long>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n }\n\n/\*\*\n \* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n @SinceKotlin("1.4")\n @kotlin.internal.InlineOnly\n public inline fun

FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Float>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n }\n\n/\*\*\n \* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n @SinceKotlin("1.4")\n @kotlin.internal.InlineOnly\n public inline fun

DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Double>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n }\n\n/\*\*\n \* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n @SinceKotlin("1.4")\n @kotlin.internal.InlineOnly\n public inline fun

BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Boolean>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n }\n\n/\*\*\n \* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n @SinceKotlin("1.4")\n @kotlin.internal.InlineOnly\n public inline fun

CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Char>(size).apply { add(accumulator)





```

runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting
list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {\n return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) ->
R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original
array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes the index of an element, current accumulator value\n * and the element itself, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic

```



```

not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun <R> CharArray.scanIndexed(initial:
R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n return runningFoldIndexed(initial,
operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Array<out T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ByteArray.sumBy(selector: (Byte) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ShortArray.sumBy(selector: (Short) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
IntArray.sumBy(selector: (Int) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince
= "1.5")\npublic inline fun LongArray.sumBy(selector: (Long) -> Int): Int {\n var sum: Int = 0\n for (element
in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
FloatArray.sumBy(selector: (Float) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
DoubleArray.sumBy(selector: (Double) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
BooleanArray.sumBy(selector: (Boolean) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharArray.sumBy(selector: (Char) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the
array.\n *\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Array<out T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by

```



```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.sumOf(selector: (Long) -> Double): Double {\n var sum: Double = 0.toDouble()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.sumOf(selector: (Double) -> Double): Double {\n var sum: Double = 0.toDouble()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Double): Double {\n var sum: Double = 0.toDouble()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.sumOf(selector: (Char) -> Double): Double {\n var sum: Double = 0.toDouble()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Array<out T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.sumOf(selector: (Byte) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sumOf(selector: (Short) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun IntArray.sumOf(selector: (Int) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector]
function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfInt")kotlin.internal.InlineOnlypublic inline fun
LongArray.sumOf(selector: (Long) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in this) {
 sum +=
selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector] function
applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfInt")kotlin.internal.InlineOnlypublic
inline fun FloatArray.sumOf(selector: (Float) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector]
function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfInt")kotlin.internal.InlineOnlypublic inline fun
DoubleArray.sumOf(selector: (Double) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector]
function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfInt")kotlin.internal.InlineOnlypublic
inline fun BooleanArray.sumOf(selector: (Boolean) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in
this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by
[selector] function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfInt")kotlin.internal.InlineOnlypublic inline fun
CharArray.sumOf(selector: (Char) -> Int): Int {
 var sum: Int = 0.toInt()
 for (element in this) {
 sum +=
selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector] function
applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfLong")kotlin.internal.InlineOnlypublic
inline fun <T> Array<out T>.sumOf(selector: (T) -> Long): Long {
 var sum: Long = 0.toLong()
 for
(element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values
produced by [selector] function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfLong")kotlin.internal.InlineOnlypublic inline fun
ByteArray.sumOf(selector: (Byte) -> Long): Long {
 var sum: Long = 0.toLong()
 for (element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector]
function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfLong")kotlin.internal.InlineOnlypublic
inline fun ShortArray.sumOf(selector: (Short) -> Long): Long {
 var sum: Long = 0.toLong()
 for (element in
this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by
[selector] function applied to each element in the array.

*SinceKotlin("1.4")OptIn(kotlin.experimental.ExperimentalTypeInference::class)OverloadResolution
ByLambdaReturnTypekotlin.jvm.JvmName("sumOfLong")kotlin.internal.InlineOnlypublic inline fun
IntArray.sumOf(selector: (Int) -> Long): Long {
 var sum: Long = 0.toLong()
 for (element in this) {
 sum += selector(element)
 }
 return sum
}
// Returns the sum of all values produced by [selector]
function applied to each element in the array.

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.sumOf(selector: (Long) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.sumOf(selector: (Double) -> Long): Long {\n var sum: Long = 0.toLong()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.sumOf(selector: (Char) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {\n var
sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n *
Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {\n var
sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n *
Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {\n var
sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each
element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {\n var
sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt
{\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt
{\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong):
ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong
{\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

```



```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {\n
var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong
{\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n *

```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong
{\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> ULong):
ULong {\n var sum: ULong = 0.toULong()\n

```

for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\*\*\n \* Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong):
ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns an original collection
containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n

```

```

*\npublic fun <T : Any> Array<T>.requireNonNulls(): Array<T> {\n for (element in this) {\n if (element ==
null) {\n throw IllegalArgumentException("null element found in $this.")\n }\n }\n

```

```

@Suppress("UNCHECKED_CAST")\n return this as Array<T>\n}\n\n/**\n * Splits the original array into pair
of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n *\npublic inline fun <T> Array<out
T>.partition(predicate:

```

```

(T) -> Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val second = ArrayList<T>()\n for
(element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n

```

```

second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of
lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n val first =
ArrayList<Byte>()\n val second = ArrayList<Byte>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded
`true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n val first =
ArrayList<Short>()\n val second = ArrayList<Short>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n val first =
ArrayList<Int>()\n val second = ArrayList<Int>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n val first =
ArrayList<Long>()\n val second = ArrayList<Long>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first*
list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which
[predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n val first =
ArrayList<Float>()\n val second = ArrayList<Float>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n val first =
ArrayList<Double>()\n val second = ArrayList<Double>()\n for (element in this) {\n if
(predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains
elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate]
yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>,
List<Boolean>> {\n val first = ArrayList<Boolean>()\n val second = ArrayList<Boolean>()\n for (element in
this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {\n val first =
ArrayList<Char>()\n val second = ArrayList<Char>()\n for (element in this) {\n if (predicate(element)) {\n
 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun

```

```

 first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first,
second)\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the
same index.\n * The returned list has length of the shortest collection.\n * \n *
@sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <T, R> Array<out
T>.zip(other: Array<out R>): List<Pair<T, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a
list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has
length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic
infix fun <R> ByteArray.zip(other: Array<out R>): List<Pair<Byte, R>> {\n return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> ShortArray.zip(other: Array<out
R>): List<Pair<Short, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun
<R> IntArray.zip(other: Array<out R>): List<Pair<Int, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> LongArray.zip(other: Array<out
R>): List<Pair<Long, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun
<R> FloatArray.zip(other: Array<out R>): List<Pair<Float, R>> {\n return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> DoubleArray.zip(other: Array<out
R>): List<Pair<Double, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R>
BooleanArray.zip(other: Array<out R>): List<Pair<Boolean, R>> {\n return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> CharArray.zip(other: Array<out R>):
List<Pair<Char, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the
elements of `this` array and the [other] array with the same index\n * using the provided [transform] function
applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <T, R, V> Array<out
T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val
list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other]
array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n
}\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>

```

```

ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V> BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

```

Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public infix fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

```

Returns a list of pairs built from the elements of `this` collection and [other] array with the same index. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterable

```

```

*public infix fun <R> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

```

Returns a list of pairs built from the elements of `this` collection and [other] array with the same index. The returned list has length of the shortest collection. @sample

```

samples.collections.Iterables.Operations.zipIterable

```



```

(a: Int, b: R) -> V): List<V> {\n val arraySize = size\n val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n
if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline
fun <R, V> LongArray.zip(other: Iterable<R>, transform: (a: Long, b: R) -> V): List<V> {\n val arraySize =
size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element
in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other]
collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V>
FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {\n val arraySize = size\n val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n
if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V>
DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {\n val arraySize = size\n val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other)
{\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic
inline fun <R, V> BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {\n val
arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10),
arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n
list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the
elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function
applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V>
CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {\n val arraySize = size\n val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n
if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a
list of pairs built from the elements of `this` array and the [other] array
with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun ByteArray.zip(other: ByteArray):
List<Pair<Byte, Byte>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun
ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun IntArray.zip(other: IntArray):
List<Pair<Int, Int>> {\n

```

```

return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and
the [other] array with the same index. The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

public infix fun LongArray.zip(other: LongArray):
List<Pair<Long, Long>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest
collection.

@sample
samples.collections.Iterables.Operations.zipIterable

public infix fun
FloatArray.zip(other: FloatArray): List<Pair<Float, Float>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The
returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

public infix fun DoubleArray.zip(other: DoubleArray):
List<Pair<Double, Double>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index. The returned list has length of the
shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

public infix fun
BooleanArray.zip(other: BooleanArray): List<Pair<Boolean, Boolean>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index. The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

public infix fun CharArray.zip(other: CharArray):
List<Pair<Char, Char>> {
 return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of values built
from the elements of `this` array and the [other] array with the same index using the provided [transform]
function applied to each pair of elements. The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

public inline fun <V> ByteArray.zip(other:
ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

Returns a list of values built from the elements of `this` array and the [other] array with the same
index using the provided [transform] function applied to each pair of elements. The returned list has length
of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

public inline fun <V> ShortArray.zip(other: ShortArray, transform: (a: Short,
b: Short) -> V): List<V> {
 val size = minOf(size, other.size)
 val list = ArrayList<V>(size)
 for (i in 0
until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

Returns a list of values built
from the elements of `this` array and the [other] array with the same index using the provided [transform]
function applied to each pair of elements. The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

public inline fun <V>
IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {
 val size = minOf(size, other.size)
 val list =
ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return
list
}

Returns a list of values built from the elements of `this` array and the [other] array with the same
index using the provided [transform] function applied
to each pair of elements. The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

public inline fun <V>
LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {
 val size = minOf(size,
other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size) {
 list.add(transform(this[i], other[i]))
 }
 return list
}

Returns a list of values built from the elements of `this` array and the [other] array
with the same index using the provided [transform] function applied to each pair of elements. The returned
list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

public inline fun <V>
FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {
 val size = minOf(size,
other.size)
 val list = ArrayList<V>(size)
 for (i in 0 until size)
{
 list.add(transform(this[i], other[i]))
 }
 return list
}

Returns a list of values built from the
elements of `this` array and the [other] array with the same index using the provided [transform] function

```

```

applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <V>
DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V):
List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <V>
CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n
}\n return list\n}\n\n/**\n * Appends the string
from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If
the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n *
elements will be appended, followed by the [truncated] string (which defaults to `"...`").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \npublic fun <T, A : Appendable> Array<out
T>.joinTo(buffer: A, separator: CharSequence = ``, ``, prefix: CharSequence = ``, postfix: CharSequence = ``,
limit: Int = -1, truncated: CharSequence = `"...`, transform: ((T) -> CharSequence)? = null): A {\n
buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n
 if (limit < 0 || count <= limit) {\n buffer.appendElement(element, transform)\n } else break\n }\n if
(limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n
return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to `"...`").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun
<A : Appendable> ByteArray.joinTo(buffer: A, separator: CharSequence = ``, ``, prefix: CharSequence = ``,
postfix: CharSequence = ``, limit: Int = -1, truncated: CharSequence = `"...`, transform: ((Byte) ->
CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count
> 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit)
buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will
be appended, followed by the [truncated] string (which defaults to `"...`").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A : Appendable>
ShortArray.joinTo(buffer: A, separator: CharSequence = ``, ``, prefix: CharSequence = ``, postfix: CharSequence
= ``, limit: Int = -1, truncated: CharSequence = `"...`, transform: ((Short) -> CharSequence)? = null): A {\n
buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n
 if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit)
buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the
elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection

```





```

null): A {
 buffer.append(prefix)
 var count = 0
 for (element in this) {
 if (++count > 1)
 buffer.append(separator)
 if (limit < 0 || count <= limit) {
 if (transform != null)
 buffer.append(transform(element))
 else
 buffer.append(element.toString())
 }
 else break
 }
 if (limit >= 0 && count > limit) buffer.append(truncated)
 buffer.append(postfix)
 return buffer
}

Appendable * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinTo

public fun <A : Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = "\", prefix: CharSequence = "\\", postfix: CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): A {
 buffer.append(prefix)
 var count = 0
 for (element in this) {
 if (++count > 1)
 buffer.append(separator)
 if (limit < 0 || count <= limit) {
 if (transform != null)
 buffer.append(transform(element))
 else
 buffer.append(element)
 }
 else break
 }
 if (limit >= 0 && count > limit) buffer.append(truncated)
 buffer.append(postfix)
 return buffer
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinToString

public fun <T> Array<out T>.joinToString(separator: CharSequence = "\", prefix: CharSequence = "\\", postfix: CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
 return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinToString

public fun ByteArray.joinToString(separator: CharSequence = "\", prefix: CharSequence = "\\", postfix: CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...", transform: ((Byte) -> CharSequence)? = null): String {
 return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinToString

public fun ShortArray.joinToString(separator: CharSequence = "\", prefix: CharSequence = "\\", postfix: CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...", transform: ((Short) -> CharSequence)? = null): String {
 return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinToString

public fun IntArray.joinToString(separator: CharSequence = "\", prefix: CharSequence = "\\", postfix: CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): String {
 return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
@sample
samples.collections.Collections.Transformations.joinToString

public fun LongArray.joinToString(separator:

```



```

being iterated.\n *\npublic fun BooleanArray.asIterable(): Iterable<Boolean> {\n if (isEmpty()) return
emptyList()\n return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original
array returning its elements when being iterated.\n */\npublic fun CharArray.asIterable(): Iterable<Char> {\n if
(isEmpty()) return emptyList()\n return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that
wraps the original array returning its elements when being iterated.\n * \n * @sample
samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun <T> Array<out T>.asSequence():
Sequence<T> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ByteArray.asSequence():
Sequence<Byte> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ShortArray.asSequence():
Sequence<Short> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun IntArray.asSequence():
Sequence<Int> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun LongArray.asSequence():
Sequence<Long> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun FloatArray.asSequence():
Sequence<Float> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun DoubleArray.asSequence():
Sequence<Double> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator()
}\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being
iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun
BooleanArray.asSequence(): Sequence<Boolean> {\n if (isEmpty()) return emptySequence()\n return Sequence
{ this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements
when being iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic
fun CharArray.asSequence(): Sequence<Char> {\n if (isEmpty()) return emptySequence()\n return Sequence {
this.iterator() }\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun Array<out Byte>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN
else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Array<out Short>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Array<out Int>.average(): Double {\n var sum: Double
= 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return if (count
== 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun Array<out Long>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Array<out Float>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/

```

```

*^@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Array<out Double>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*^public fun ByteArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in
this) {\n sum += element\n ++count\n }\n return if (count ==
0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*^public fun ShortArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n ++count\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the array.\n
*^public fun IntArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n ++count\n }\n return if (count == 0)
Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*^public fun LongArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n ++count\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the array.\n
*^public fun FloatArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n ++count\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the array.\n
*^public fun DoubleArray.average(): Double {\n var sum: Double =
0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return if (count
== 0) Double.NaN else sum / count\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*^@kotlin.jvm.JvmName("sumOfByte")\npublic fun Array<out Byte>.sum(): Int {\n var sum: Int = 0\n for
(element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*^@kotlin.jvm.JvmName("sumOfShort")\npublic fun Array<out Short>.sum(): Int {\n var sum: Int =
0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n
* Returns the sum of all elements in the array.\n
*^@kotlin.jvm.JvmName("sumOfInt")\npublic fun Array<out
Int>.sum(): Int {\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n return
sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*^@kotlin.jvm.JvmName("sumOfLong")\npublic fun Array<out Long>.sum(): Long {\n var sum: Long = 0L\n
for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in
the array.\n
*^@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Array<out Float>.sum(): Float {\n var sum:
Float = 0.0f\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of
all elements in the array.\n
*^@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Array<out Double>.sum():
Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += element\n }\n return
sum\n}\n\n/**\n * Returns
the sum of all elements in the array.\n
*^public fun ByteArray.sum(): Int {\n var sum: Int = 0\n for (element in
this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*^public fun ShortArray.sum(): Int {\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*^public fun IntArray.sum(): Int {\n var
sum: Int = 0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of
all elements in the array.\n
*^public fun LongArray.sum(): Long {\n var sum: Long = 0L\n for (element in this)
{\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*^public fun FloatArray.sum(): Float {\n var sum: Float = 0.0f\n for (element in this) {\n sum += element\n }\n
return sum\n}\n\n/**\n * Returns
the sum of all elements in the array.\n
*^public fun DoubleArray.sum(): Double {\n var sum: Double = 0.0\n
for (element in this) {\n sum += element\n }\n return sum\n}\n\n"/*\n * Copyright 2010-2022 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*^@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("RangesKt")\n\npackage
kotlin.ranges\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:

```

<https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib>\n\nimport kotlin.random.\*\n\n \* Returns the first element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun IntProgression.first(): Int {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.first\n}\n\n \* Returns the first element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun LongProgression.first(): Long {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.first\n}\n\n \* Returns the first element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun CharProgression.first(): Char {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.first\n}\n\n \* Returns the first element, or `null` if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun IntProgression.firstOrNull(): Int? {\n return if (isEmpty()) null else this.first\n}\n\n \* Returns the first element, or `null` if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun LongProgression.firstOrNull(): Long? {\n return if (isEmpty()) null else this.first\n}\n\n \* Returns the first element, or `null` if the progression is empty.\n\n \* \n\n@SinceKotlin("1.7")\npublic fun CharProgression.firstOrNull(): Char? {\n return if (isEmpty()) null else this.first\n}\n\n \* Returns the last element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun IntProgression.last(): Int {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.last\n}\n\n \* Returns the last element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun LongProgression.last(): Long {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.last\n}\n\n \* Returns the last element.\n \* \n \* @throws NoSuchElementException if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun CharProgression.last(): Char {\n if (isEmpty())\n throw NoSuchElementException("Progression \$this is empty.")\n return this.last\n}\n\n \* Returns the last element, or `null` if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun IntProgression.lastOrNull(): Int? {\n return if (isEmpty()) null else this.last\n}\n\n \* Returns the last element, or `null` if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun LongProgression.lastOrNull(): Long? {\n return if (isEmpty()) null else this.last\n}\n\n \* Returns the last element, or `null` if the progression is empty.\n \* \n \* @sample samples.collections.Collections.Elements.last\n\n \* \n\n@SinceKotlin("1.7")\npublic fun CharProgression.lastOrNull(): Char? {\n return if (isEmpty()) null else this.last\n}\n\n \* Returns a random element from this range.\n \* \n \* @throws IllegalArgumentException if this range is empty.\n\n \* \n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun IntRange.random(): Int {\n return random(Random)\n}\n\n \* Returns a random element from this range.\n \* \n \* @throws IllegalArgumentException if this range is empty.\n\n \* \n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun LongRange.random(): Long {\n return random(Random)\n}\n\n \* Returns a random element from this range.\n \* \n \* @throws IllegalArgumentException if this range is empty.\n\n \* \n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharRange.random(): Char {\n return random(Random)\n}\n\n \* Returns a random element from this range using the specified source of randomness.\n \* \n \* @throws IllegalArgumentException if this range is empty.\n\n \* \n\n@SinceKotlin("1.3")\npublic fun IntRange.random(random: Random): Int {\n try {\n return random.nextInt(this)\n } catch (e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n \* Returns a random element from this range using the specified source of randomness.\n \* \n \* @throws

```

IllegalArgumentException if this range is empty.\n *\n@SinceKotlin("1.3")\npublic fun
LongRange.random(random: Random): Long {\n try {\n return random.nextLong(this)\n } catch(e:
IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness.\n * \n * @throws
IllegalArgumentException if this range is empty.\n *\n@SinceKotlin("1.3")\npublic fun
CharRange.random(random: Random): Char {\n try {\n return random.nextInt(first.code, last.code +
1).toChar()\n } catch(e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n
}\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun IntRange.randomOrNull(): Int? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongRange.randomOrNull(): Long? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharRange.randomOrNull(): Char? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
fun IntRange.randomOrNull(random: Random): Int? {\n if (isEmpty())\n return null\n return
random.nextInt(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongRange.randomOrNull(random: Random): Long? {\n if (isEmpty())\n return null\n return
random.nextLong(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharRange.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return
random.nextInt(first.code, last.code + 1).toChar()\n}\n\n/**\n * Returns `true` if this range contains the specified
[element].\n * \n * Always
returns `false` if the [element] is `null`.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline
operator fun IntRange.contains(element: Int?): Boolean {\n return element != null &&
contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns
`false` if the [element] is `null`.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator
fun LongRange.contains(element: Long?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n *
Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is
`null`.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun
CharRange.contains(element: Char?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n *\n@kotlin.jvm.JvmName("intRangeContains")\npublic
operator fun ClosedRange<Int>.contains(value:
Byte): Boolean {\n return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n *\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun
ClosedRange<Long>.contains(value: Byte): Boolean {\n return contains(value.toLong())\n}\n\n/**\n * Checks if
the specified [value] belongs to this range.\n *\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator
fun ClosedRange<Short>.contains(value: Byte): Boolean {\n return contains(value.toShort())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n *\n@Deprecated("This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun

```

```

ClosedRange<Double>.contains(value: Byte): Boolean {
 return contains(value.toDouble())
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value: Byte): Boolean {
 return contains(value.toFloat())
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.jvm.JvmName("intRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Byte): Boolean {
 return contains(value.toInt())
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.jvm.JvmName("longRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Byte): Boolean {
 return contains(value.toLong())
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.jvm.JvmName("shortRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Short>.contains(value: Byte): Boolean {
 return contains(value.toShort())
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Byte): Boolean {
 return (this as ClosedRange<Int>).contains(value)
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Byte): Boolean {
 return (this as ClosedRange<Long>).contains(value)
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Double): Boolean {
 return value.toIntExactOrNull().let { if (it != null) contains(it) else false }
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Double): Boolean {
 return value.toLongExactOrNull().let { if (it != null) contains(it) else false }
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Double): Boolean {
 return value.toByteExactOrNull().let { if (it != null) contains(it) else false }
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value: Double): Boolean {
 return value.toShortExactOrNull().let { if (it != null) contains(it) else false }
}

* Checks if the specified [value] belongs to this range.
*\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value: Double): Boolean {
 return contains(value.toFloat())
}

* Checks if the specified [value] belongs to this range.
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Float): Boolean {
 return value.toIntExactOrNull().let { if (it != null) contains(it) else false }
}

```



```

Checks if the specified [value] belongs to this range.\n *^@Deprecated("This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value:
Float): Boolean {\n return value.toLongExactOrNull().let
 { if (it != null) contains(it) else false } }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Float): Boolean {\n return value.toByteExactOrNull().let { if (it != null)
contains(it) else false } }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun
ClosedRange<Short>.contains(value: Float): Boolean {\n return value.toShortExactOrNull().let
 { if (it != null) contains(it) else false } }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun ClosedRange<Double>.contains(value:
Float): Boolean {\n return contains(value.toDouble()) }\n\n/**\n * Checks if the specified [value] belongs to this
range.\n
*^@kotlin.jvm.JvmName("doubleRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Double>.contains(value: Float): Boolean {\n return
contains(value.toDouble()) }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Int):
Boolean {\n return contains(value.toLong()) }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Int):
Boolean {\n
 return value.toByteExactOrNull().let { if (it != null) contains(it) else false } }\n\n/**\n * Checks if the specified
[value] belongs to this range.\n
*^@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun
ClosedRange<Short>.contains(value: Int): Boolean {\n return value.toShortExactOrNull().let { if (it != null)
contains(it) else false } }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Int): Boolean {\n return contains(value.toDouble()) }\n\n/**\n * Checks
if the specified [value] belongs to this range.\n
*^@Deprecated("This `contains` operation mixing integer and
floating point arguments has ambiguous
semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4",
hiddenSince = "1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun
ClosedRange<Float>.contains(value: Int): Boolean {\n return contains(value.toFloat()) }\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("longRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Long>.contains(value: Int): Boolean {\n return contains(value.toLong()) }\n\n/**\n
* Checks if the specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("byteRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Byte>.contains(value: Int): Boolean {\n return value.toByteExactOrNull().let { if (it
!= null) contains(it) else false } }\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*^@kotlin.jvm.JvmName("shortRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic

```



removed.})\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Short): Boolean {\n return contains(value.toFloat())\n}\n\n/\*\*\n \* Checks if the specified [value] belongs to this range.\n \*/\n\n\*\n@kotlin.jvm.JvmName(\"intRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Short): Boolean {\n return contains(value.toInt())\n}\n\n/\*\*\n \* Checks if the specified [value] belongs to this range.\n \*/\n\n\*\n@kotlin.jvm.JvmName(\"longRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Short): Boolean {\n return contains(value.toLong())\n}\n\n/\*\*\n \* Checks if the specified [value] belongs to this range.\n \*/\n\n\*\n@kotlin.jvm.JvmName(\"byteRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Byte>.contains(value: Short): Boolean {\n return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/\*\*\n \* Checks if the specified [value] belongs to this range.\n \*/\n\n\*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Short): Boolean {\n return (this as ClosedRange<Int>).contains(value)\n}\n\n/\*\*\n \* Checks if the specified [value] belongs to this range.\n \*/\n\n\*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Short): Boolean {\n return (this as ClosedRange<Long>).contains(value)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Int.downTo(to: Byte): IntProgression {\n return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Long.downTo(to: Byte): LongProgression {\n return LongProgression.fromClosedRange(this, to.toLong(), -1L)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Byte.downTo(to: Byte): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Short.downTo(to: Byte): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Char.downTo(to: Char): CharProgression {\n return CharProgression.fromClosedRange(this, to, -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Int.downTo(to: Int): IntProgression {\n return IntProgression.fromClosedRange(this, to, -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Long.downTo(to: Int): LongProgression {\n return LongProgression.fromClosedRange(this, to.toLong(), -1L)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Byte.downTo(to: Int): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to, -1)\n}\n\n/\*\*\n \* Returns a progression from this value down to the specified [to] value with the step -1.\n \* \n \* The [to] value should be less than or equal to `this` value.\n \* If the [to] value is greater than `this` value the returned progression is empty.\n \*/\n\n\*\npublic infix fun Short.downTo(to: Int): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to, -1)\n}\n\n/\*\*\n \* Returns a progression

from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Int.downTo(to: Long): LongProgression {
 return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Long.downTo(to: Long): LongProgression {
 return LongProgression.fromClosedRange(this, to, -1L)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Byte.downTo(to: Long): LongProgression {
 return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Short.downTo(to: Long): LongProgression {
 return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Int.downTo(to: Short): IntProgression {
 return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Byte.downTo(to: Short): IntProgression {
 return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

```

Returns a progression from this value down to the specified [to] value with the step -1. The [to] value should be less than or equal to this value. If the [to] value is greater than this value the returned progression is empty.

```

public infix fun Short.downTo(to: Short): IntProgression {
 return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

```

Returns a progression that goes over the same range in the opposite direction with the same step.

```

public fun IntProgression.reversed(): IntProgression {
 return IntProgression.fromClosedRange(last, first, -step)
}

```

Returns a progression that goes over the same range in the opposite direction with the same step.

```

public fun LongProgression.reversed(): LongProgression {
 return LongProgression.fromClosedRange(last, first, -step)
}

```

Returns a progression that goes over the same range in the opposite direction with the same step.

```

public fun CharProgression.reversed(): CharProgression {
 return CharProgression.fromClosedRange(last, first, -step)
}

```

Returns a progression that goes over the same range with the given step.

```

public infix fun IntProgression.step(step: Int): IntProgression {
 checkStepIsPositive(step > 0, step)
 return IntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

```

Returns a progression that goes over the same range with the given step.

```

public infix fun LongProgression.step(step: Long): LongProgression {
 checkStepIsPositive(step > 0, step)
 return LongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

```

Returns a progression that goes over the same range with the given step.

```

public infix fun CharProgression.step(step: Int): CharProgression {
 checkStepIsPositive(step > 0, step)
 return CharProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

```

internal fun Int.toByteExactOrNull(): Byte? {
 return if (this in Byte.MIN\_VALUE.toInt()..Byte.MAX\_VALUE.toInt()) this.toByte() else null
}

internal fun Long.toByteExactOrNull(): Byte? {
 return if (this in Byte.MIN\_VALUE.toLong()..Byte.MAX\_VALUE.toLong()) this.toByte() else null
}

internal fun Short.toByteExactOrNull(): Byte? {
 return if (this in

```

Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort()
this.toByte() else null\n}\n\ninternal fun Double.toByteExactOrNull(): Byte? {\n return if (this in
Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toInt().toByte() else null\n}\n\ninternal fun
Float.toByteExactOrNull(): Byte? {\n return if (this in
Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat()) this.toInt().toByte() else null\n}\n\ninternal fun
Long.toIntExactOrNull(): Int? {\n return if (this in Int.MIN_VALUE.toLong()..Int.MAX_VALUE.toLong())
this.toInt() else null\n}\n\ninternal fun Double.toIntExactOrNull(): Int? {\n return if (this in
Int.MIN_VALUE.toDouble()..Int.MAX_VALUE.toDouble()) this.toInt() else null\n}\n\ninternal fun
Float.toIntExactOrNull(): Int? {\n return if (this in Int.MIN_VALUE.toFloat()..Int.MAX_VALUE.toFloat())
this.toInt() else null\n}\n\ninternal fun Double.toLongExactOrNull(): Long? {\n return if (this in
Long.MIN_VALUE.toDouble()..Long.MAX_VALUE.toDouble()) this.toLong() else null\n}\n\ninternal
fun Float.toLongExactOrNull(): Long? {\n return if (this in
Long.MIN_VALUE.toFloat()..Long.MAX_VALUE.toFloat()) this.toLong() else null\n}\n\ninternal fun
Int.toShortExactOrNull(): Short? {\n return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())
this.toShort() else null\n}\n\ninternal fun Long.toShortExactOrNull(): Short? {\n return if (this in
Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong()) this.toShort() else null\n}\n\ninternal fun
Double.toShortExactOrNull(): Short? {\n return if (this in
Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble()) this.toInt().toShort() else null\n}\n\ninternal fun
Float.toShortExactOrNull(): Short? {\n return if (this in
Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat()) this.toInt().toShort() else null\n}\n\n**\n * Returns
a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to
`this` value, then the returned range is empty.\n */\npublic infix
fun Int.until(to: Byte): IntRange {\n return this .. (to.toInt() - 1).toInt()\n}\n\n**\n * Returns a range from this
value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then
the returned range is empty.\n */\npublic infix fun Long.until(to: Byte): LongRange {\n return this .. (to.toLong() -
1).toLong()\n}\n\n**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If
the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
Byte): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n**\n * Returns a range from this value up to
but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Short.until(to: Byte): IntRange {\n return this.toInt() .. (to.toInt() -
1).toInt()\n}\n\n**\n * Returns
a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to
`this` value, then the returned range is empty.\n */\npublic infix fun Char.until(to: Char): CharRange {\n if (to <=
'\u0000') return CharRange.EMPTY\n return this .. (to - 1).toChar()\n}\n\n**\n * Returns a range from this value
up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the
returned range is empty.\n */\npublic infix fun Int.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return
IntRange.EMPTY\n return this .. (to - 1).toInt()\n}\n\n**\n * Returns a range from this value up to but excluding
the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is
empty.\n */\npublic infix fun Long.until(to: Int): LongRange {\n return this .. (to.toLong() -
1).toLong()\n}\n\n**\n * Returns a range from this value up to
but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Byte.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return
IntRange.EMPTY\n return this.toInt() .. (to - 1).toInt()\n}\n\n**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Short.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return
IntRange.EMPTY\n return this.toInt() .. (to - 1).toInt()\n}\n\n**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Int.until(to: Long): LongRange {\n if (to <= Long.MIN_VALUE) return

```

LongRange.EMPTY\n return this.toLong() .. (to - 1).toLong()\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Long.until(to: Long): LongRange {\n if (to <= Long.MIN\_VALUE) return LongRange.EMPTY\n return this .. (to - 1).toLong()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Byte.until(to: Long): LongRange {\n if (to <= Long.MIN\_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to - 1).toLong()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Short.until(to: Long): LongRange {\n if (to <= Long.MIN\_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to - 1).toLong()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Int.until(to: Short): IntRange {\n return this .. (to.toInt() - 1).toInt()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Long.until(to: Short): LongRange {\n return this .. (to.toLong() - 1).toLong()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Byte.until(to: Short): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/\*\*\n \* Returns a range from this value up to but excluding the specified [to] value.\n \* \n \* If the [to] value is less than or equal to `this` value, then the returned range is empty.\n \*/\npublic infix fun Short.until(to: Short): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeastComparable\n \*/\npublic fun <T : Comparable<T>> T.coerceAtLeast(minimumValue: T): T {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Byte.coerceAtLeast(minimumValue: Byte): Byte {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Short.coerceAtLeast(minimumValue: Short): Short {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Int.coerceAtLeast(minimumValue: Int): Int {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Long.coerceAtLeast(minimumValue: Long): Long {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Float.coerceAtLeast(minimumValue: Float): Float {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value is not less than the specified [minimumValue].\n \* \n \* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n \* \n \* @sample samples.comparisons.ComparableOps.coerceAtLeast\n \*/\npublic fun Double.coerceAtLeast(minimumValue: Double): Double {\n return if (this < minimumValue) minimumValue else this\n}\n\n/\*\*\n \* Ensures that this value

```

is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the
[maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostComparable\n * \n public fun <T : Comparable<T>>
T.coerceAtMost(maximumValue: T): T {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less
than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun Byte.coerceAtMost(maximumValue: Byte):
Byte {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not
greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the
[maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun Short.coerceAtMost(maximumValue: Short): Short {\n return if (this > maximumValue)
maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n *
@return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n *
@sample samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun Int.coerceAtMost(maximumValue:
Int): Int {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not
greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the
[maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun Long.coerceAtMost(maximumValue: Long):
Long {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not
greater than the specified [maximumValue].\n * \n *
@return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n *
@sample samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun
Float.coerceAtMost(maximumValue: Float): Float {\n return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMost\n * \n public fun Double.coerceAtMost(maximumValue:
Double): Double {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this
value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range,
or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n * \n public fun <T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n if
(minimumValue != null && maximumValue != null) {\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this >
maximumValue) return maximumValue\n }\n else {\n if (minimumValue != null && this <
minimumValue) return minimumValue\n if (maximumValue != null && this > maximumValue) return
maximumValue\n }\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n * \n public
fun Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {\n if (minimumValue > maximumValue)
throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n * \n public fun Short.coerceIn(minimumValue: Short,

```





```

@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.
 * @sample samples.comparisons.ComparableOps.coerceIn
 *
public fun Int.coerceIn(range: ClosedRange<Int>): Int {
 if (range is
 ClosedFloatingPointRange) {
 return this.coerceIn<Int>(range)
 }
 if (range.isEmpty()) throw
 IllegalArgumentException("Cannot coerce value to an empty range: $range.")
 return when {
 this < range.start -> range.start
 this > range.endInclusive -> range.endInclusive
 else -> this
 }
}
 *
Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if
this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 *
 * @sample samples.comparisons.ComparableOps.coerceIn
 *
public fun Long.coerceIn(range:
 ClosedRange<Long>): Long {
 if (range is ClosedFloatingPointRange) {
 return
 this.coerceIn<Long>(range)
 }
 if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value
 to an empty range: $range.")
 return when {
 this < range.start -> range.start
 this >
 range.endInclusive
 else -> this
 }
}
 *
 * Copyright 2010-2023 JetBrains s.r.o. and Kotlin
Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.
 *
package kotlin
import
kotlin.experimental
import
kotlin.jvm
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@JvmInline
public value class UByte @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal
 constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {
 companion object {
 /**
 * A constant holding the minimum value an instance of UByte can have.
 */
 public const val
 MIN_VALUE: UByte = UByte(0)
 /**
 * A constant holding the maximum value an instance of
 UByte can have.
 */
 public const val
 MAX_VALUE: UByte = UByte(-1)
 /**
 * The number of bytes used to represent an instance of UByte in a binary form.
 */
 public const val
 SIZE_BYTES: Int = 1
 /**
 * The number of bits used to represent an instance of
 UByte in a binary form.
 */
 public const val
 SIZE_BITS: Int = 8
 }
 /**
 * Compares this
 value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a
 negative number if it's less than other,
 * or a positive number if it's greater than other.
 */
 @kotlin.internal.InlineOnly
 @Suppress("OVERRIDE_BY_INLINE")
 public override inline operator fun
 compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())
 /**
 * Compares this value with the
 specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if
 it's less than other,
 * or a positive number if it's
 greater than other.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun compareTo(other: UShort):
 Int = this.toInt().compareTo(other.toInt())
 /**
 * Compares this value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if
 it's less than other,
 * or a positive number if it's greater than other.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)
 /**
 * Compares this value with the
 specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if
 it's less than other,
 * or a positive number if it's greater than other.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)
 /**
 * Adds the
 other value to this value.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())
 /**
 * Adds the other
 value to this value.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun plus(other: UShort): UInt =
 this.toUInt().plus(other.toUInt())
 /**
 * Adds the other value to this value.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)
 /**
 * Adds the other value to this
 value.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun plus(other: ULong): ULong =
 this.toULong().plus(other)
 /**
 * Subtracts the other value from this value.
 */
 @kotlin.internal.InlineOnly
 public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())
 /**
 * Subtracts the

```

```

other value from this value. */n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())n /**
Subtracts the other value from this value. */n @kotlin.internal.InlineOnly\n public inline operator fun
minus(other: UInt): UInt = this.toUInt().minus(other)n /** Subtracts the other value from this value. */n
@kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong =
this.toULong().minus(other)n\n /** Multiplies this value by the other value. */n @kotlin.internal.InlineOnly\n
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())n /** Multiplies this
value by the other value. */n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt
= this.toUInt().times(other.toUInt())n /** Multiplies this value by the other value. */n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)n
/** Multiplies this value by the other value. */n @kotlin.internal.InlineOnly\n public inline operator
fun times(other: ULong): ULong = this.toULong().times(other)n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. */n @kotlin.internal.InlineOnly\n public inline operator fun div(other:
ULong): ULong = this.toULong().div(other)n\n /**n
* Calculates the remainder of truncating division of this value by the other value.n * \n * The result is
always less than the divisor.n */n @kotlin.internal.InlineOnly\n public inline operator fun rem(other:
UByte): UInt = this.toUInt().rem(other.toUInt())n /**n * Calculates the remainder of truncating division of
this value by the other value.n * \n * The result is always less than the divisor.n */n
@kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): UInt =
this.toUInt().rem(other.toUInt())n /**n * Calculates the remainder of truncating division of this value by the
other value.n * \n * The result is always less than the divisor.n */n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)n /**n * Calculates the remainder of
truncating division of this value by the other value.n * \n * The result is always
less than the divisor.n */n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong):
ULong = this.toULong().rem(other)n\n /**n * Divides this value by the other value, flooring the result to an
integer that is closer to negative infinity.n * \n * For unsigned types, the results of flooring division and
truncating division are the same.n */n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UByte): UInt = this.toUInt().floorDiv(other.toUInt())n /**n * Divides this value by the other value, flooring
the result to an integer that is closer to negative infinity.n * \n * For unsigned types, the results of flooring
division and truncating division are the same.n */n @kotlin.internal.InlineOnly\n public inline fun
floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())n /**n * Divides this value by the other
value, flooring the result to an integer that is closer
to negative infinity.n * \n * For unsigned types, the results of flooring division and truncating division are the
same.n */n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): UInt =
this.toUInt().floorDiv(other)n /**n * Divides this value by the other value, flooring the result to an integer that
is closer to negative infinity.n * \n * For unsigned types, the results of flooring division and truncating division
are the same.n */n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong =
this.toULong().floorDiv(other)n\n /**n * Calculates the remainder of flooring division of this value by the
other value.n * \n * The result is always less than the divisor.n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.n */n @kotlin.internal.InlineOnly\n public inline
fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()n

```

```

/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is
always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating
division are the same.\n * \n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort =
this.toUInt().mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n * \n @kotlin.internal.InlineOnly\n
public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n /**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n
* For unsigned
types, the remainders of flooring division and truncating division are the same.\n * \n
@kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n
/**\n * Returns this value incremented by one.\n * \n * @sample samples.misc.Builtins.inc\n * \n
@kotlin.internal.InlineOnly\n public inline operator fun inc(): UByte = UByte(data.inc())\n\n /**\n * Returns
this value decremented by one.\n * \n * @sample samples.misc.Builtins.dec\n * \n
@kotlin.internal.InlineOnly\n public inline operator fun dec(): UByte = UByte(data.dec())\n\n /** Creates a
range from this value to the specified [other] value. * \n @kotlin.internal.InlineOnly\n public inline operator fun
rangeTo(other: UByte): UIntRange = UIntRange(this.toUInt(), other.toUInt())\n\n /**\n * Creates a range from
this value up to but excluding the specified [other] value.\n * \n * If the [other] value is less
than or equal to `this` value, then the returned range is empty.\n * \n @SinceKotlin("1.7")\n
@ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: UByte):
UIntRange = this.toUInt() until other.toUInt()\n\n /** Performs a bitwise AND operation between the two values.
* \n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UByte): UByte = UByte(this.data and
other.data)\n\n /** Performs a bitwise OR operation between the two values. * \n @kotlin.internal.InlineOnly\n
public inline infix fun or(other: UByte): UByte = UByte(this.data or other.data)\n\n /** Performs a bitwise XOR
operation between the two values. * \n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UByte):
UByte = UByte(this.data xor other.data)\n\n /** Inverts the bits in this value. * \n @kotlin.internal.InlineOnly\n
public inline fun inv(): UByte = UByte(data.inv())\n\n /**\n * Converts this
[UByte] value to [Byte].\n * \n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting
`Byte` value represents\n * the same numerical value as this `UByte`. Otherwise the result is negative.\n * \n
* The resulting `Byte` value has the same binary representation as this `UByte` value.\n * \n
@kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data\n\n /**\n * Converts this [UByte] value to
[Short].\n * \n * The resulting `Short` value represents the same numerical value as this `UByte`. \n * \n *
The least significant 8 bits of the resulting `Short` value are the same as the bits of this `UByte` value,\n *
whereas the most significant 8 bits are filled with zeros.\n * \n @kotlin.internal.InlineOnly\n public inline fun
toShort(): Short = data.toShort() and 0xFF\n\n /**\n * Converts this [UByte] value to [Int].\n * \n * The
resulting `Int` value represents the same numerical value as
this `UByte`.\n * \n * The least significant 8 bits of the resulting `Int` value are the same as the bits of this
`UByte` value,\n * whereas the most significant 24 bits are filled with zeros.\n * \n
@kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data.toInt() and 0xFF\n\n /**\n * Converts this
[UByte] value to [Long].\n * \n * The resulting `Long` value represents the same numerical value as this
`UByte`.\n * \n * The least significant 8 bits of the resulting `Long` value are the same as the bits of this
`UByte` value,\n * whereas the most significant 56 bits are filled with zeros.\n * \n
@kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFF\n\n /** Returns this
value. * \n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = this\n\n /**\n * Converts this
[UByte] value to [UShort].\n * \n * The resulting `UShort` value represents the same numerical
value as this `UByte`.\n * \n * The least significant 8 bits of the resulting `UShort` value are the same as the
bits of this `UByte` value,\n * whereas the most significant 8 bits are filled with zeros.\n * \n
@kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = UShort(data.toShort() and 0xFF)\n\n /**\n

```

```

* Converts this [UByte] value to [UInt].\n *\n * The resulting `UInt` value represents the same numerical value
as this `UByte`.\n *\n * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this
`UByte` value,\n * whereas the most significant 24 bits are filled with zeros.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFF)\n /**\n *
Converts this [UByte] value to [ULong].\n *\n * The resulting `ULong` value represents the same numerical
value as this `UByte`.\n *\n * The least significant 8 bits of the resulting
`ULong` value are the same as the bits of this `UByte` value,\n * whereas the most significant 56 bits are filled
with zeros.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong()
and 0xFF)\n /**\n * Converts this [UByte] value to [Float].\n *\n * The resulting `Float` value represents
the same numerical value as this `UByte`.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toFloat():
Float = this.toInt().toFloat()\n /**\n * Converts this [UByte] value to [Double].\n *\n * The resulting
`Double` value represents the same numerical value as this `UByte`.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = this.toInt().toDouble()\n public override fun toString(): String =
toInt().toString()\n }\n /**\n * Converts this [Byte] value to [UByte].\n *\n * If this value is positive, the resulting
`UByte` value represents the same numerical value as
this `Byte`.\n *\n * The resulting `UByte` value has the same binary representation as this `Byte` value.\n
*/\n
@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun Byte.toUByte(): UByte = UByte(this)\n /**\n * Converts this [Short] value to [UByte].\n *\n * If
this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n *
the same numerical value as this `Short`.\n *\n * The resulting `UByte` value is represented by the least significant 8
bits of this `Short` value.\n
*/\n
@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun Short.toUByte(): UByte = UByte(this.toByte())\n /**\n * Converts this [Int] value to [UByte].\n
*\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value
represents\n * the same numerical value as this `Int`.\n *\n * The resulting `UByte`
value is represented by the least significant 8 bits of this `Int` value.\n
*/\n
@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun Int.toUByte(): UByte = UByte(this.toByte())\n /**\n * Converts this [Long] value to [UByte].\n
*\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value
represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UByte` value is represented by the
least significant 8 bits of this `Long` value.\n
*/\n
@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun Long.toUByte(): UByte = UByte(this.toByte())\n ,"/*\n * Copyright 2010-2023 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n // Auto-generated file. DO NOT EDIT!\n \npackage
kotlin\n \nimport kotlin.experimental.*\n \nimport
kotlin.jvm.*\n \n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\n
public value class UInt @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi
internal val data: Int) : Comparable<UInt> {\n companion object {\n /**\n * A constant holding the
minimum value an instance of UInt can have.\n */\n public const val MIN_VALUE: UInt = UInt(0)\n }\n
 /**\n * A constant holding the maximum value an instance of UInt can have.\n */\n public const val
MAX_VALUE: UInt = UInt(-1)\n /**\n * The number of bytes used to represent an instance of UInt in a
binary form.\n */\n public const val SIZE_BYTES: Int = 4\n /**\n * The number of bits used to
represent an instance of UInt in a binary form.\n */\n public const val SIZE_BITS: Int = 32\n }\n
 /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified
other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n
@kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UByte): Int =

```

```

this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
compareTo(other: UShort): Int = this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive number if it's greater than
other.\n */\n @kotlin.internal.InlineOnly\n @Suppress(\"OVERRIDE_BY_INLINE\")\n public override
inline operator fun compareTo(other: UInt): Int = uintCompare(this.data, other.data)\n\n /**\n * Compares this
value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a
negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n
@kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: ULong): Int =
this.toULong().compareTo(other)\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())\n\n /** Adds the other value to this
value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UShort): UInt =
this.plus(other.toUInt())\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))\n\n /** Adds the other value to
this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong =
this.toULong().plus(other)\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UByte): UInt = this.minus(other.toUInt())\n\n /** Subtracts the other value
from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): UInt =
this.minus(other.toUInt())\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UInt): UInt = UInt(this.data.minus(other.data))\n\n /** Subtracts the other
value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong
= this.toULong().minus(other)\n\n /** Multiplies this value
by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt =
this.times(other.toUInt())\n\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun times(other: UShort): UInt = this.times(other.toUInt())\n\n /** Multiplies this value by the
other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt =
UInt(this.data.times(other.data))\n\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong =
this.toULong().times(other)\n\n /** Divides this value by the other value, truncating the result to an integer that is
closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UByte): UInt =
this.div(other.toUInt())\n\n /** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UShort): UInt =
this.div(other.toUInt())\n\n /** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UInt): UInt = uintDivide(this,
other)\n\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n
@kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())\n\n /**\n * Calculates the remainder of
truncating division of this value by the other value.\n * \n * The result
is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other:
UShort): UInt = this.rem(other.toUInt())\n\n /**\n * Calculates the remainder of truncating division of this value
by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun rem(other: UInt): UInt = uintRemainder(this, other)\n\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong): ULong =

```

this.toULong().rem(other)\n\n /\*\*\n \* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \* \n \* For unsigned types, the results of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n

public inline fun floorDiv(other: UByte): UInt = this.floorDiv(other.toUInt())\n /\*\*\n \* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \* \n \* For unsigned types, the results of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n

public inline fun floorDiv(other: UShort): UInt = this.floorDiv(other.toUInt())\n /\*\*\n \* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \* \n \* For unsigned types, the results of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n

public inline fun floorDiv(other: UInt): UInt = div(other)\n /\*\*\n \* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \* \n \* For unsigned types, the results of flooring division and

truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n

@kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.mod(other.toUInt()).toUByte()\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating

division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort = this.mod(other.toUInt()).toUShort()\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n

@kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = rem(other)\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n

@kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /\*\*\n \* Returns this value incremented by one.\n \* \n \* @sample

samples.misc.Builtins.inc\n \*/\n @kotlin.internal.InlineOnly\n public inline operator fun inc(): UInt = UInt(data.inc())\n\n /\*\*\n \* Returns this value decremented by one.\n \* \n \* @sample

samples.misc.Builtins.dec\n \*/\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UInt = UInt(data.dec())\n\n /\*\*\n \* Creates a range from this value to the specified [other] value.\n \*/\n

@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UInt): UIntRange = UIntRange(this, other)\n\n /\*\*\n \* Creates a range from this value up to but excluding the specified [other] value.\n \* \n \* If the [other] value is less than or equal to `this` value, then the returned range is empty.\n \*/\n

@SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: UInt): UIntRange = this.until(other)\n\n /\*\*\n \* Shifts this value left by the [bitCount] number of bits.\n \* \n \* Note that only the five lowest-order bits of the [bitCount] are used as the shift distance.\n \* \n \* The shift distance actually used is therefore always in the range `0..31`.\n \*/\n

@kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount: Int): UInt = UInt(data shl bitCount)\n\n /\*\*\n \* Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros.\n \* \n \* Note that only the five lowest-order bits of the [bitCount] are used as the shift distance.\n \* \n \* The shift distance actually used is therefore always in the range `0..31`.\n \*/\n @kotlin.internal.InlineOnly\n

public inline infix fun shr(bitCount: Int): UInt = UInt(data ushr bitCount)\n\n /\*\*\n \* Performs a bitwise AND operation between the two values.\n \*/\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UInt):

UInt = UInt(this.data and other.data)\n\n /\*\*\n \* Performs a bitwise OR operation between the two values.\n \*/\n

@kotlin.internal.InlineOnly\n public inline infix fun or(other: UInt): UInt = UInt(this.data or other.data)\n\n /\*\*\n \* Performs a bitwise XOR operation between the two values.\n \*/\n

```

@kotlin.internal.InlineOnly\n public inline infix fun xor(other: UInt): UInt = UInt(this.data xor other.data)\n /**
Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n public inline fun inv(): UInt =
UInt(data.inv())\n\n /**\n * Converts this [UInt] value to [Byte].\n *\n * If this value is less than or equals
to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this `UInt`.\n
*\n * The resulting `Byte` value is represented by the least significant 8 bits of this `UInt` value.\n * Note that
the resulting `Byte` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte():
Byte = data.toByte()\n /**\n * Converts this [UInt] value to [Short].\n *\n * If this value is less than or
equals to [Short.MAX_VALUE], the resulting `Short` value represents\n * the same numerical value as this
`UInt`.\n *\n * The resulting `Short` value is represented
by the least significant 16 bits of this `UInt` value.\n * Note that the resulting `Short` value may be negative.\n
*/\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data.toShort()\n /**\n * Converts this
[UInt] value to [Int].\n *\n * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int` value
represents\n * the same numerical value as this `UInt`. Otherwise the result is negative.\n *\n * The resulting
`Int` value has the same binary representation as this `UInt` value.\n */\n @kotlin.internal.InlineOnly\n public
inline fun toInt(): Int = data\n /**\n * Converts this [UInt] value to [Long].\n *\n * The resulting `Long`
value represents the same numerical value as this `UInt`.\n *\n * The least significant 32 bits of the resulting
`Long` value are the same as the bits of this `UInt` value,\n * whereas the most significant 32 bits are filled with
zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFFFF_FFFF\n\n
/**\n * Converts this [UInt] value to [UByte].\n *\n * If this value is less than or equals to
[UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `UInt`.\n
*\n * The resulting `UByte` value is represented by the least significant 8 bits of this `UInt` value.\n */\n
 @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = data.toUByte()\n /**\n * Converts this
[UInt] value to [UShort].\n *\n * If this value is less than or equals to [UShort.MAX_VALUE], the resulting
`UShort` value represents\n * the same numerical value as this `UInt`.\n *\n * The resulting `UShort` value is
represented by the least significant 16 bits of this `UInt` value.\n */\n @kotlin.internal.InlineOnly\n public
inline fun toUShort(): UShort = data.toUShort()\n /** Returns
this value. */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = this\n /**\n * Converts this
[UInt] value to [ULong].\n *\n * The resulting `ULong` value represents the same numerical value as this
`UInt`.\n *\n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `UInt`
value,\n * whereas the most significant 32 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n
public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF_FFFF)\n /**\n * Converts this
[UInt] value to [Float].\n *\n * The resulting value is the closest `Float` to this `UInt` value.\n * In case when
this `UInt` value is exactly between two `Float`s,\n * the one with zero at least significant bit of mantissa is
selected.\n */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toDouble().toFloat()\n
/**\n * Converts this [UInt] value to
[Double].\n *\n * The resulting `Double` value represents the same numerical value as this `UInt`.\n */\n
 @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = uintToDouble(data)\n\n public override
fun toString(): String = toLong().toString()\n\n /**\n * Converts this [Byte] value to [UInt].\n *\n * If
this value is positive, the resulting `UInt` value represents the same numerical value as this `Byte`.\n *\n * The
least significant 8 bits of the resulting `UInt` value are the same as the bits of this `Byte` value,\n * whereas the
most significant 24 bits are filled with the sign bit of this value.\n */\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Byte.toUInt(): UInt = UInt(this.toInt())\n /**\n * Converts this [Short] value to [UInt].\n *\n
* If this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`.\n *\n
* The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value,\n *
whereas the most significant 16 bits are filled with the sign bit of this value.\n */

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toUInt(): UInt = UInt(this.toInt())\n/**\n * Converts this [Int] value to [UInt].\n *\n * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Int`. \n *\n * The resulting `UInt` value has the same binary representation as this `Int` value.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toUInt(): UInt = UInt(this)\n/**\n * Converts this [Long] value to [UInt].\n *\n * If this value is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents\n * the same numerical value as this `Long`. \n *\n * The resulting `UInt` value is represented by the least significant 32 bits of this `Long` value.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUInt(): UInt = UInt(this.toDouble())\n/**\n * Converts this [Float] value to [UInt].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toUInt(): UInt = doubleToUInt(this.toDouble())\n/**\n * Converts this [Double] value to [UInt].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Double` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`. \n
\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic\ninline fun Double.toUInt(): UInt = doubleToUInt(this)\n"/\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport kotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UShort @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal\nconstructor(@PublishedApi internal val data: Short) : Comparable<UShort> {\n\n companion object {\n\n /**\n * A constant holding the minimum value an instance of UShort can have.\n */\n public const val MIN_VALUE: UShort = UShort(0)\n\n /**\n * A constant holding the maximum value an instance of UShort can have.\n */\n public const val MAX_VALUE: UShort = UShort(-1)\n\n /**\n * The number of bytes used to represent an instance of UShort in a binary form.\n */\n public const val SIZE_BYTES: Int = 2\n\n /**\n * The number of bits used to represent an instance of UShort in a binary form.\n */\n public const val SIZE_BITS: Int = 16\n }\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public override inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n /**\n * Adds the other value to this value.\n */\n @kotlin.internal.InlineOnly\n
```



```

 public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())\n /** Adds the other
value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UShort): UInt =
this.toUInt().plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)\n /** Adds the other value to this
value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong =
this.toULong().plus(other)\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n /** Subtracts the
other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())\n
 /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun
minus(other: UInt): UInt = this.toUInt().minus(other)\n /** Subtracts the other value from this value. */\n
 @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong =
this.toULong().minus(other)\n\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())\n /** Multiplies this
value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt
= this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. */\n
 @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n
 /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator
fun times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)\n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other:
ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is
always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other:
UByte): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of
this value by the other value.\n * \n * The result is always less than the divisor.\n */\n
 @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): UInt =
this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the
other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)\n /**\n * Calculates the remainder of
truncating division of this value by the other value.\n * \n * The result is always
less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong):
ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by the other value, flooring the result to an
integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and
truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring
the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring
division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun
floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other
value, flooring the result to an integer that is closer
to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the
same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): UInt =
this.toUInt().floorDiv(other)\n /**\n * Divides this value by the other value, flooring the result to an integer that

```

is closer to negative infinity.\n \* \n \* For unsigned types, the results of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()\n\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()\n\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n\n /\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is always less than the divisor.\n \* \n \* For unsigned types, the remainders of flooring division and truncating division are the same.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /\*\*\n \* Returns this value incremented by one.\n \* \n \* @sample samples.misc.Builtins.inc\n \*/\n @kotlin.internal.InlineOnly\n public inline operator fun inc(): UShort = UShort(data.inc())\n\n /\*\*\n \* Returns this value decremented by one.\n \* \n \* @sample samples.misc.Builtins.dec\n \*/\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UShort = UShort(data.dec())\n\n /\*\* Creates a range from this value to the specified [other] value. \*/\n @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UShort): UIntRange = UIntRange(this.toUInt(), other.toUInt())\n\n /\*\* Creates a range from this value up to but excluding the specified [other] value. \n \* \n \* If the [other] value is less than or equal to `this` value, then the returned range is empty.\n \*/\n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: UShort): UIntRange = this.toUInt() until other.toUInt()\n\n /\*\* Performs a bitwise AND operation between the two values. \*/\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UShort): UShort = UShort(this.data and other.data)\n\n /\*\* Performs a bitwise OR operation between the two values. \*/\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: UShort): UShort = UShort(this.data or other.data)\n\n /\*\* Performs a bitwise XOR operation between the two values. \*/\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UShort): UShort = UShort(this.data xor other.data)\n\n /\*\* Inverts the bits in this value. \*/\n @kotlin.internal.InlineOnly\n public inline fun inv(): UShort = UShort(data.inv())\n\n /\*\* Converts this [UShort] value to [Byte]. \n \* \n \* If this value is less than or equals to [Byte.MAX\_VALUE], the resulting `Byte` value represents the same numerical value as this `UShort`. \n \* \n \* The resulting `Byte` value is represented by the least significant 8 bits of this `UShort` value. \n \* \n \* Note that the resulting `Byte` value may be negative.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n\n /\*\* Converts this [UShort] value to [Short]. \n \* \n \* If this value is less than or equals to [Short.MAX\_VALUE], the resulting `Short` value represents the same numerical value as this `UShort`. \n \* \n \* Otherwise the result is negative. \n \* \n \* The resulting `Short` value has the same binary representation as this `UShort` value.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data\n\n /\*\* Converts this [UShort] value to [Int]. \n \* \n \* The resulting `Int` value represents the same numerical value as this `UShort`. \n \* \n \* The least significant 16 bits of the resulting `Int` value are the same as the bits of this `UShort` value, \n \* \n \* whereas the most significant 16 bits are filled with zeros.\n \*/\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data.toInt()\n\n /\*\* Converts this [UShort] value to [Long]. \n \* \n \* The resulting `Long` value represents the same numerical value as this `UShort`. \n \* \n \* The least significant 16 bits of the resulting `Long` value are the same as the bits of this `UShort` value, \n \* \n \* whereas the most significant 48 bits are filled

```

with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and
0xFFFF\n\n /**\n * Converts this [UShort] value to [UByte].\n * If this value is less than or equals to
[UByte.MAX_VALUE], the resulting `UByte` value represents\n
 * the same numerical value as this `UShort`.\n * The resulting `UByte` value is represented by the least
significant 8 bits of this `UShort` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUByte():
UByte = data.toUByte()\n /** Returns this value. */\n @kotlin.internal.InlineOnly\n public inline fun
toUShort(): UShort = this\n /**\n * Converts this [UShort] value to [UInt].\n * The resulting `UInt`
value represents the same numerical value as this `UShort`.\n * The least significant 16 bits of the resulting
`UInt` value are the same as the bits of this `UShort` value,\n * whereas the most significant 16 bits are filled with
zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF)\n
 /**\n * Converts this [UShort] value to [ULong].\n * The resulting `ULong` value represents the same
numerical value as this `UShort`.\n
 * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `UShort`
value,\n * whereas the most significant 48 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n
public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF)\n\n /**\n * Converts this [UShort]
value to [Float].\n * The resulting `Float` value represents the same numerical value as this `UShort`.\n
 */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toInt().toFloat()\n /**\n *
Converts this [UShort] value to [Double].\n * The resulting `Double` value represents the same numerical
value as this `UShort`.\n */\n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double =
this.toInt().toDouble()\n\n public override fun toString(): String = toInt().toString()\n\n /**\n * Converts this
[Byte] value to [UShort].\n * If this value is positive,
 the resulting `UShort` value represents the same numerical value as this `Byte`.\n * The least significant 8 bits
of the resulting `UShort` value are the same as the bits of this `Byte` value,\n * whereas the most significant 8 bits
are filled with the sign bit of this value.\n
 */\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Byte.toUShort(): UShort = UShort(this.toShort())\n\n /**\n * Converts this [Short] value to
[UShort].\n * If this value is positive, the resulting `UShort` value represents the same numerical value as this
`Short`.\n * The resulting `UShort` value has the same binary representation as this `Short` value.\n
 */\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Short.toUShort(): UShort = UShort(this)\n\n /**\n * Converts this [Int] value to [UShort].\n * If
this value is positive and less than or equals to [UShort.MAX_VALUE],
 the resulting `UShort` value represents\n * the same numerical value as this `Int`.\n * The resulting `UShort`
value is represented by the least significant 16 bits of this `Int` value.\n
 */\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Int.toUShort(): UShort = UShort(this.toShort())\n\n /**\n * Converts this [Long] value to
[UShort].\n * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort`
value represents\n * the same numerical value as this `Long`.\n * The resulting `UShort` value is represented by
the least significant 16 bits of this `Long` value.\n
 */\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Long.toUShort(): UShort = UShort(this.toShort())\n\n /*\n * Copyright 2010-2021 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source
 code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
 */\n\n @file:kotlin.jvm.JvmMultifileClass\n @file:kotlin.jvm.JvmName("CollectionsKt")\n @file:OptIn(kotlin.exper
imental.ExperimentalTypeInference::class)\n @package kotlin.collections\n @import kotlin.contracts.*\n @import
kotlin.random.Random\n @internal object EmptyIterator : ListIterator<Nothing> {\n override fun hasNext():
Boolean = false\n override fun hasPrevious(): Boolean = false\n override fun nextIndex(): Int = 0\n override
fun previousIndex(): Int = -1\n override fun next(): Nothing = throw NoSuchElementException()\n override fun
previous(): Nothing = throw NoSuchElementException()\n }\n @internal object EmptyList : List<Nothing>,

```

```

Serializable, RandomAccess {
 private const val serialVersionUID: Long = -7390468764508069838L
 override fun equals(other: Any?): Boolean = other is List<*> && other.isEmpty()
 override fun hashCode(): Int = 1
 override fun toString(): String = ""
 override val size: Int get() = 0
 override fun isEmpty(): Boolean = true
 override fun contains(element: Nothing): Boolean = false
 override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()
 override fun get(index: Int): Nothing = throw IndexOutOfBoundsException("Empty list doesn't contain element at index $index.")
 override fun indexOf(element: Nothing): Int = -1
 override fun lastIndexOf(element: Nothing): Int = -1
 override fun iterator(): Iterator<Nothing> = EmptyIterator
 override fun listIterator(): ListIterator<Nothing> = EmptyIterator
 override fun listIterator(index: Int): ListIterator<Nothing> {
 if (index != 0) throw IndexOutOfBoundsException("Index: $index")
 return EmptyIterator
 }
 override fun subList(fromIndex: Int, toIndex: Int): List<Nothing> {
 if (fromIndex == 0 && toIndex == 0) return this
 throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex")
 }
 private fun readResolve(): Any = EmptyList
}

internal fun <T> Array<out T>.asCollection(): Collection<T> = ArrayAsCollection(this, isVarargs = false)

private class ArrayAsCollection<T>(val values: Array<out T>, val isVarargs: Boolean) : Collection<T> {
 override val size: Int get() = values.size
 override fun isEmpty(): Boolean = values.isEmpty()
 override fun contains(element: T): Boolean = values.contains(element)
 override fun containsAll(elements: Collection<T>): Boolean = elements.all { contains(it) }
 override fun iterator(): Iterator<T> = values.iterator()
 // override hidden toArray implementation to prevent copying of values array
 public fun toArray(): Array<out Any?> = values.copyOfToArrayOfAny(isVarargs)
}

/**
 * Returns an empty read-only list. The returned list is serializable (JVM).
 */
@sample samples.collections.Collections.Lists.emptyReadOnlyList

public fun <T> emptyList(): List<T> = EmptyList

/**
 * Returns a new read-only list of given elements. The returned list is serializable (JVM).
 */
@sample samples.collections.Collections.Lists.readOnlyList

public fun <T> listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()

/**
 * Returns an empty read-only list. The returned list is serializable (JVM).
 */
@sample samples.collections.Collections.Lists.emptyReadOnlyList

public inline fun <T> listOf(): List<T> = emptyList()

/**
 * Returns an empty new [MutableList].
 */
@sample samples.collections.Collections.Lists.emptyMutableList

public inline fun <T> mutableListOf(): MutableList<T> = ArrayList()

/**
 * Returns an empty new [ArrayList].
 */
@sample samples.collections.Collections.Lists.emptyArrayList

public inline fun <T> arrayListOf(): ArrayList<T> = ArrayList()

/**
 * Returns a new [MutableList] with the given elements.
 */
@sample samples.collections.Collections.Lists.mutableList

public fun <T> mutableListOf(vararg elements: T): MutableList<T> = if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))

/**
 * Returns a new [ArrayList] with the given elements.
 */
@sample samples.collections.Collections.Lists.arrayList

public fun <T> arrayListOf(vararg elements: T): ArrayList<T> = if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))

/**
 * Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The returned list is serializable (JVM).
 */
@sample samples.collections.Collections.Lists.listOfNotNull

public fun <T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()

/**
 * Returns a new read-only list only of those given elements, that are not null. The returned list is serializable (JVM).
 */
@sample samples.collections.Collections.Lists.listOfNotNull

public fun <T : Any> listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()

/**
 * Creates a new read-only list with the specified [size], where each element is calculated by calling the specified [init] function.
 */
@sample samples.collections.Collections.Lists.listOf

public fun <T> listOf(size: Int, init: () -> T): List<T> = List(size, init)

/**
 * The function [init] is called for each list element sequentially starting from the first one.
 */
@sample samples.collections.Collections.Lists.listOf

public fun <T> listOf(size: Int, init: () -> T): List<T> = List(size, init)

```

```

index.\n *\n * @sample samples.collections.Collections.Lists.readOnlyListFromInitializer\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> List(size: Int, init: (index: Int) -> T):
List<T> = MutableList(size, init)\n\n/**\n * Creates a new mutable list with the specified [size], where each element
is calculated by calling the specified\n * [init] function.\n *\n * The function [init]
is called for each list element sequentially starting from the first one.\n * It should return the value for a list element
given its index.\n *\n * @sample samples.collections.Collections.Lists.mutableListFromInitializer\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> MutableList(size: Int, init: (index:
Int) -> T): MutableList<T> {\n val list = ArrayList<T>(size)\n repeat(size) { index -> list.add(init(index)) }\n
return list}\n\n/**\n * Builds a new read-only [List] by populating a [MutableList] using the given
[builderAction]\n * and returning a read-only list with the same elements.\n *\n * The list passed as a receiver to the
[builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified
behavior.\n *\n * The returned list is serializable (JVM).\n *\n * @sample
samples.collections.Builders.Lists.buildListSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic
inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>().->Unit): List<E> {\n contract {
callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n return
buildListInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninter
nal expect inline fun <E> buildListInternal(builderAction: MutableList<E>().->Unit): List<E>\n\n/**\n * Builds a
new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list
with the same elements.\n *\n * The list passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * The returned list is
serializable (JVM).\n *\n * [capacity] is used to hint the expected number of elements added in the
[builderAction].\n *\n * @throws IllegalArgumentException if the given [capacity] is negative.\n
*\n * @sample samples.collections.Builders.Lists.buildListSampleWithCapacity\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <E> buildList(capacity: Int, @BuilderInference builderAction:
MutableList<E>().->Unit): List<E> {\n contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }\n return buildListInternal(capacity,
builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>().->Unit): List<E>\n\n/**\n * Returns an
[IntRange] of the valid indices for this collection.\n *\n * @sample
samples.collections.Collections.Collections.indicesOfCollection\n *\npublic val Collection<*>.indices: IntRange\n
get() = 0..size - 1\n\n/**\n * Returns the index of the last item in the list or -1 if the list is empty.\n *\n * @sample
samples.collections.Collections.Lists.lastIndexOfList\n
*\npublic val <T> List<T>.lastIndex: Int\n get() = this.size - 1\n\n/**\n * Returns `true` if the collection is not
empty.\n *\n * @sample samples.collections.Collections.Collections.collectionIsNotEmpty\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.isEmpty(): Boolean =
!isEmpty()\n\n/**\n * Returns `true` if this nullable collection is either null or empty.\n *\n * @sample
samples.collections.Collections.Collections.collectionIsNullOrEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>?.isNullOrEmpty():
Boolean {\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n return this == null ||
this.isEmpty()\n}\n\n/**\n * Returns this Collection if it's not `null` and the empty list otherwise.\n *\n * @sample
samples.collections.Collections.Collections.collectionOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Collection<T>?.orElse():
Collection<T> = this ?: emptyList()\n\n/**\n * Returns this List if it's not `null` and the empty list otherwise.\n *\n
* @sample samples.collections.Collections.Lists.listOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
<T> List<T>?.orElse(): List<T> = this ?: emptyList()\n\n/**\n * Returns this collection if it's not empty\n * or the

```

```

result of calling [defaultValue] function if the collection is empty.\n *\n * @sample
samples.collections.Collections.Collections.collectionIsEmpty\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.isEmpty(defaultValue: () ->
R): R where C : Collection<*>, C : R =\n if (isEmpty()) defaultValue() else this\n\n\n/**\n * Checks if all
elements in the specified collection are contained in this collection.\n *\n * Allows to overcome type-safety
restriction of `containsAll` that requires to pass a collection of type `Collection<E>`.\n *\n * @sample
samples.collections.Collections.Collections.collectionContainsAll\n
*/\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T>
Collection<T>.containsAll(elements: Collection<T>): Boolean = this.containsAll(elements)\n\n\n/**\n * Returns a
new list with the elements of this list randomly shuffled\n * using the specified [random] instance as the source of
randomness.\n *\n */\n@SinceKotlin("1.3")\npublic fun <T> Iterable<T>.shuffled(random: Random): List<T> =
toMutableList().apply { shuffle(random) }\n\n\ninternal fun <T> List<T>.optimizeReadOnlyList() = when (size) {\n
 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this\n}\n\n\n/**\n * Searches this list or its range for the provided
[element] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to
the Comparable natural ordering of its elements,\n * otherwise the result is undefined.\n *\n * If the list contains
multiple
elements equal to the specified [element], there is no guarantee which one will be found.\n *\n * `null` value is
considered to be less than any non-null value.\n *\n * @return the index of the element, if it is contained in the list
within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point
is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list)
still remains sorted.\n *\n * @sample samples.collections.Collections.Lists.binarySearchOnComparable\n * @sample
samples.collections.Collections.Lists.binarySearchWithBoundaries\n *\n */\npublic fun <T : Comparable<T>>
List<T?>.binarySearch(element: T?, fromIndex: Int = 0, toIndex: Int = size): Int {\n rangeCheck(size, fromIndex,
toIndex)\n\n var low = fromIndex\n var high = toIndex - 1\n\n while (low <= high) {\n val mid = (low +
high).ushr(1) // safe from overflows\n val midVal = get(mid)\n\n val cmp = compareValues(midVal, element)\n\n if (cmp < 0)\n low = mid + 1\n else if (cmp >
0)\n high = mid - 1\n else\n return mid // key found\n }\n return -(low + 1) // key not
found\n}\n\n\n/**\n * Searches this list or its range for the provided [element] using the binary search algorithm.\n *
The list is expected to be sorted into ascending order according to the specified [comparator],\n * otherwise the
result is undefined.\n *\n * If the list contains multiple elements equal to the specified [element], there is no
guarantee which one will be found.\n *\n * `null` value is considered to be less than any non-null value.\n *\n *
@return the index of the element, if it is contained in the list within the specified range;\n * otherwise, the inverted
insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be
inserted,\n * so that the list (or the specified
subrange of list) still remains sorted according to the specified [comparator].\n *\n * @sample
samples.collections.Collections.Lists.binarySearchWithComparator\n *\n */\npublic fun <T>
List<T>.binarySearch(element: T, comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Int {\n
 rangeCheck(size, fromIndex, toIndex)\n\n var low = fromIndex\n var high = toIndex - 1\n\n while (low <=
high) {\n val mid = (low + high).ushr(1) // safe from overflows\n val midVal = get(mid)\n val cmp =
comparator.compare(midVal, element)\n\n if (cmp < 0)\n low = mid + 1\n else if (cmp > 0)\n
 high = mid - 1\n else\n return mid // key found\n }\n return -(low + 1) // key not found\n}\n\n\n/**\n * Searches this list or its range for an element having the key returned by the specified [selector] function\n * equal to
the provided [key] value using the binary search algorithm.\n * The list is expected to be sorted
into ascending order according to the Comparable natural ordering of keys of its elements.\n * otherwise the result
is undefined.\n *\n * If the list contains multiple elements with the specified [key], there is no guarantee which one
will be found.\n *\n * `null` value is considered to be less than any non-null value.\n *\n * @return the index of the
element with the specified [key], if it is contained in the list within the specified range;\n * otherwise, the inverted

```

```

insertion point `(-insertion point - 1)`.n * The insertion point is defined as the index at which the element should be
inserted,n * so that the list (or the specified subrange of list) still remains sorted.n * @sample
samples.collections.Collections.Lists.binarySearchByKey\n * ^\npublic inline fun <T, K : Comparable<K>>
List<T>.binarySearchBy(\n key: K?,\n fromIndex: Int = 0,\n toIndex: Int = size,\n crossinline selector: (T) ->
K?)\n): Int =\n binarySearch(fromIndex, toIndex) { compareValues(selector(it),
key) }\n\n// do not introduce this overload --- too rare\n//public fun <T, K> List<T>.binarySearchBy(key: K,
comparator: Comparator<K>, fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =\n\n//
binarySearch(fromIndex, toIndex) { comparator.compare(selector(it), key) }\n\n\n/**\n * Searches this list or its
range for an element for which the given [comparison] function returns zero using the binary search algorithm.\n *
* The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list
elements,\n * i.e. negative values come before zero and zeroes come before positive values.\n * Otherwise, the result
is undefined.\n * \n * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee
which one will be found.\n * \n * @param comparison function that returns zero when called on the list element
being searched.\n * On the elements coming before the target element, the function
must return negative values;\n * on the elements coming after the target element, the function must return positive
values.\n * \n * @return the index of the found element, if it is contained in the list within the specified range;\n *
otherwise, the inverted insertion point `(-insertion point - 1)`.n * The insertion point is defined as the index at which
the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n *
@sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n * ^\npublic fun <T>
List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {\n rangeCheck(size,
fromIndex, toIndex)\n\n var low = fromIndex\n var high = toIndex - 1\n\n while (low <= high) {\n val mid
= (low + high).ushr(1) // safe from overflows\n val midVal = get(mid)\n val cmp = comparison(midVal)\n\n if (cmp < 0)\n low = mid + 1\n else if (cmp
> 0)\n high = mid - 1\n else\n return mid // key found\n }\n return -(low + 1) // key not
found\n}\n\n\n/**\n * Checks that `from` and `to` are in\n * the range of [0..size] and throws an appropriate exception,
if they aren't.\n * \n * @private fun rangeCheck(size: Int, fromIndex: Int, toIndex: Int) {\n when {\n fromIndex >
toIndex -> throw IllegalArgumentException("\nfromIndex ($fromIndex) is greater than toIndex ($toIndex).")\n fromIndex < 0 -> throw IndexOutOfBoundsException("\nfromIndex ($fromIndex) is less than zero.")\n toIndex
> size -> throw IndexOutOfBoundsException("\ntoIndex ($toIndex) is greater than size ($size).")\n }\n}\n\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal expect fun checkIndexOverflow(index: Int):
Int\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal expect fun checkCountOverflow(count: Int):
Int\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun throwIndexOverflow() { throw
ArithmeticException("\nIndex
overflow has happened.") }\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun throwCountOverflow() {
throw ArithmeticException("\nCount overflow has happened.") }\n\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n
*\n * @file:kotlin.jvm.JvmMultifileClass\n * @file:kotlin.jvm.JvmName("\nMapsKt")\n * @file:OptIn(kotlin.experiment
al.ExperimentalTypeInference::class)\n * @package kotlin.collections\n * @import kotlin.contracts.*\n * @private object
EmptyMap : Map<Any?, Nothing>, Serializable {\n private const val serialVersionUID: Long =
8246714829545688274\n\n override fun equals(other: Any?): Boolean = other is Map<*, *> &&
other.isEmpty()\n override fun hashCode(): Int = 0\n override fun toString(): String = "{}"\n\n override val
size: Int get() = 0\n override fun isEmpty(): Boolean = true\n\n override
fun containsKey(key: Any?): Boolean = false\n override fun containsValue(value: Nothing): Boolean = false\n\n override fun get(key: Any?): Nothing? = null\n override val entries: Set<Map.Entry<Any?, Nothing>> get() =
EmptySet\n override val keys: Set<Any?> get() = EmptySet\n override val values: Collection<Nothing> get() =
EmptyList\n\n private fun readResolve(): Any = EmptyMap\n}\n\n\n/**\n * Returns an empty read-only map of
specified type.\n * \n * The returned map is serializable (JVM).\n * \n * @sample

```

```

samples.collections.Maps.Instantiation.emptyReadOnlyMap\n *\npublic fun <K, V> emptyMap(): Map<K, V> =
@Suppress(\\"UNCHECKED_CAST\\") (EmptyMap as Map<K, V>)\n\n/**\n * Returns a new read-only map with
the specified contents, given as a list of pairs\n * where the first value is the key and the second is the value.\n *\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n *\n * Entries of the map are iterated in
the order they were specified.\n *\n * The returned map is serializable (JVM).\n *\n * @sample
samples.collections.Maps.Instantiation.mapFromPairs\n *\npublic fun <K, V> mapOf(vararg pairs: Pair<K, V>):
Map<K, V> =\n if (pairs.size > 0) pairs.toMap(LinkedHashMap(mapCapacity(pairs.size))) else
emptyMap()\n\n/**\n * Returns an empty read-only map.\n *\n * The returned map is serializable (JVM).\n *\n * @sample
samples.collections.Maps.Instantiation.emptyReadOnlyMap\n *\n@kotlin.internal.InlineOnly\npublic
inline fun <K, V> mapOf(): Map<K, V> = emptyMap()\n\n/**\n * Returns an empty new [MutableMap].\n *\n * The returned map preserves the entry iteration order.\n *\n * @sample
samples.collections.Maps.Instantiation.emptyMutableMap\n
*\n@SinceKotlin(\\"1.1\\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> mutableMapOf():
MutableMap<K, V> = LinkedHashMap()\n\n/**\n * Returns a new [MutableMap] with the specified contents, given
as a list of pairs\n * where the first component is the key
and the second is the value.\n *\n * If multiple pairs have the same key, the resulting map will contain the value
from the last of those pairs.\n *\n * Entries of the map are iterated in the order they were specified.\n *\n * @sample
samples.collections.Maps.Instantiation.mutableMapFromPairs\n * @sample
samples.collections.Maps.Instantiation.emptyMutableMap\n *\npublic fun <K, V> mutableMapOf(vararg pairs:
Pair<K, V>): MutableMap<K, V> =\n LinkedHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs)
}\n\n/**\n * Returns an empty new [HashMap].\n *\n * @sample
samples.collections.Maps.Instantiation.emptyHashMap\n
*\n@SinceKotlin(\\"1.1\\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> hashMapOf(): HashMap<K, V>
= HashMap<K, V>()\n\n/**\n * Returns a new [HashMap] with the specified contents, given as a list of pairs\n *
where the first component is the key and the second is the value.\n *\n * @sample
samples.collections.Maps.Instantiation.hashMapFromPairs\n *\npublic
fun <K, V> hashMapOf(vararg pairs: Pair<K, V>): HashMap<K, V> = HashMap<K,
V>(mapCapacity(pairs.size)).apply { putAll(pairs) }\n\n/**\n * Returns an empty new [LinkedHashMap].\n
*\n * @sample
samples.collections.Maps.Instantiation.linkedMapFromPairs\n *\n@SinceKotlin(\\"1.1\\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> linkedMapOf():
LinkedHashMap<K, V> = LinkedHashMap<K, V>()\n\n/**\n * Returns a new [LinkedHashMap] with the specified
contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n *\n * If
multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n *\n * Entries of the map are iterated in the order they were specified.\n *\n * @sample
samples.collections.Maps.Instantiation.linkedMapFromPairs\n *\npublic fun <K, V> linkedMapOf(vararg pairs:
Pair<K, V>): LinkedHashMap<K, V> = pairs.toMap(LinkedHashMap(mapCapacity(pairs.size)))\n\n/**\n * Builds
a new read-only [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning
a read-only map with the same key-value pairs.\n *\n * The map passed as a receiver to the [builderAction] is valid
only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * Entries of
the map are iterated in the order they were added by the [builderAction].\n *\n * The returned map is serializable
(JVM).\n *\n * @sample
samples.collections.Builders.Maps.buildMapSample\n
*\n@SinceKotlin(\\"1.6\\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress(\\"DEPRECATION\\")\npublic inline fun <K, V> buildMap(@BuilderInference builderAction:
MutableMap<K, V>().-> Unit): Map<K, V> {\n contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }\n return
buildMapInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\\"1.3\\")\n@kotlin.internal.InlineOnly\ninter
nal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>().-> Unit): Map<K,
V>{\n\n/**\n * Builds a new read-only

```



[Map] by populating a [MutableMap] using the given [builderAction] and returning a read-only map with the same key-value pairs. The map passed as a receiver to the [builderAction] is valid only inside that function. Using it outside of the function produces an unspecified behavior. [capacity] is used to hint the expected number of pairs added in the [builderAction]. Entries of the map are iterated in the order they were added by the [builderAction]. The returned map is serializable (JVM). @throws IllegalArgumentException if the given [capacity] is negative. @sample

```

samples.collections.Builders.Maps.buildMapSample
*/
@SinceKotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION")
public inline fun <K, V> buildMap(capacity: Int, @BuilderInference builderAction:
MutableMap<K, V>().() -> Unit): Map<K, V> {
 contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE)
 }
 return buildMapInternal(capacity,
builderAction)
}
@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal expect inline
fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>().() -> Unit): Map<K, V>
Calculate the initial capacity of a map.
@PublishedApi
internal expect fun mapCapacity(expectedSize: Int):
Int
Returns `true` if this map is not empty. @sample
samples.collections.Maps.Usage.mapIsNotEmpty
*/
@kotlin.internal.InlineOnly
public inline fun <K, V>
Map<out K, V>.isEmpty(): Boolean = !isNotEmpty()
Returns `true` if this nullable map is either null or
empty. @sample
samples.collections.Maps.Usage.mapIsNullOrEmpty
*/
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K,
V>?.isEmpty(): Boolean {
 contract {
 returns(false) implies (this@isEmpty != null)
 }
 return this == null || isEmpty()
}
Returns the [Map] if its not `null`, or the empty [Map] otherwise. @sample
samples.collections.Maps.Usage.mapOrElse
*/
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<K,
V>?.orElse(): Map<K, V> = this ?: emptyMap()
Returns this map if it's not empty or the result of
calling [defaultValue] function if the map is empty. @sample
samples.collections.Maps.Usage.mapIfEmpty
*/
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public
inline fun <M, R> M.ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R =
 if (isEmpty())
 defaultValue() else this
Checks if the map contains the given key. This method allows to use the
`x in map` syntax for checking whether an object is contained in the map. @sample
samples.collections.Maps.Usage.containsKey
*/
@kotlin.internal.InlineOnly
public inline operator fun
<@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.contains(key: K): Boolean =
containsKey(key)
Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.
*/
@kotlin.internal.InlineOnly
public inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K,
V>.get(key: K): V? =
 @Suppress("UNCHECKED_CAST") (this as Map<K, V>).get(key)
Allows to use the index operator for storing values in a mutable map.
*/
@kotlin.internal.InlineOnly
public inline
operator fun <K, V> MutableMap<K, V>.set(key: K, value: V): Unit {
 put(key, value)
}
Returns `true` if the map contains the specified [key]. Allows to overcome type-safety restriction of `containsKey`
that requires to pass a key of type `K`.
*/
@kotlin.internal.InlineOnly
public inline fun
<@kotlin.internal.OnlyInputTypes K> Map<out K, *>.containsKey(key: K): Boolean =
@Suppress("UNCHECKED_CAST") (this as Map<K, *>).containsKey(key)
Returns `true` if the map
maps one or more keys to the specified [value].
*/
Allows to overcome type-safety restriction of `containsValue` that requires to pass a value of type `V`.
@sample
samples.collections.Maps.Usage.containsValue
*/
@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases
@kotlin.internal.InlineOnly
public inline fun <K, @kotlin.internal.OnlyInputTypes V> Map<K,
V>.containsValue(value: V): Boolean = this.containsValue(value)
Removes the specified key and its
corresponding value from this map. @return the previous value associated with the key, or `null` if the key

```

was not present in the map.

\* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

```

@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes K, V>
MutableMap<out K, V>.remove(key: K): V? =
 @Suppress("UNCHECKED_CAST") (this as MutableMap<K, V>).remove(key)

```

\* Returns the key component of the map entry.

\* This method allows to use destructuring declarations when working with maps, for example:

```

for ((key, value) in map) {
 // do something with the key and the value
}

```

\* Returns the value component of the map entry.

\* This method allows to use destructuring declarations when working with maps, for example:

```

for ((key, value) in map) {
 // do something with the key and the value
}

```

\* Returns the value component of the map entry.

\* Converts entry to [Pair] with key being first component and value being second.

```

@kotlin.internal.InlineOnly
public inline fun <K, V> Map.Entry<K, V>.toPair(): Pair<K, V> = Pair(key, value)

```

\* Returns the value for the given [key] if the value is present and not `null`. Otherwise, returns the result of the [defaultValue] function.

\* @sample samples.collections.Maps.Usage.getOrNull

```

@kotlin.internal.InlineOnly
public inline fun <K, V> Map<K, V>.getOrNull(key: K, defaultValue: () -> V): V =
 get(key) ?: defaultValue

```

\* Returns the value for the given [key] if the value is present and not `null`. Otherwise, returns the result of the [defaultValue] function.

\* @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default value was provided for that map.

```

@kotlin.internal.InlineOnly
public fun <K, V> Map<K, V>.getValue(key: K): V =
 getOrImplicitDefault(key)

```

\* Returns the value for the given [key] if the value is present and not `null`. Otherwise, calls the [defaultValue] function, puts its result into the map under the given key and returns the call result.

\* Note that the operation is not guaranteed to be atomic if the map is being modified concurrently.

\* @sample samples.collections.Maps.Usage.getOrPut

```

@kotlin.internal.InlineOnly
public inline fun <K, V> MutableMap<K, V>.getOrPut(key: K, defaultValue: () -> V): V {
 val value = get(key)
 return if (value == null) {
 val answer = defaultValue()
 put(key, answer)
 answer
 } else {
 value
 }
}

```

\* Returns an [Iterator] over the entries in the [Map].

\* @sample samples.collections.Maps.Usage.forOverEntries

```

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.iterator(): Iterator<Map.Entry<K, V>> = entries.iterator()

```

\* Returns a [MutableIterator] over the mutable entries in the [MutableMap].

```

@kotlin.jvm.JvmName("mutableIterator")
@kotlin.internal.InlineOnly
public inline operator fun <K, V> MutableMap<K, V>.iterator(): MutableIterator<MutableMap.MutableEntry<K, V>> = entries.iterator()

```

\* Populates the given [destination] map with entries having the keys of this map and the values obtained by applying the [transform] function to each entry in this [Map].

```

public inline fun <K, V, R, M : MutableMap<in K, in R>> Map<out K, V>.mapValuesTo(destination: M, transform: (Map.Entry<K, V>) -> R): M {
 return entries.associateByTo(destination, { it.key }, transform)
}

```

\* Populates the given [destination] map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.

\* In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

```

public inline fun <K, V, R, M : MutableMap<in R, in V>> Map<out K, V>.mapKeysTo(destination: M, transform: (Map.Entry<K, V>) -> R): M {
 return entries.associateByTo(destination, transform, { it.value })
}

```

\* Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.

```

public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Array<out Pair<K, V>>): Unit {
 for ((key, value) in pairs) {
 put(key, value)
 }
}

```

\* Puts all the elements of the given collection into

```

this [MutableMap] with the first component in the pair being the key and the second the value.\n *\npublic fun <K,
V> MutableMap<in K, in V>.putAll(pairs: Iterable<Pair<K, V>>): Unit {\n for ((key, value) in pairs) {\n
put(key, value)\n }\n}\n\n/**\n * Puts all the elements of the given sequence into this [MutableMap] with the first
component in the pair being the key and the second the value.\n *\npublic fun <K, V> MutableMap<in
K, in V>.putAll(pairs: Sequence<Pair<K, V>>): Unit {\n for ((key, value) in pairs) {\n put(key, value)\n
}\n}\n\n/**\n * Returns a new map with entries having the keys of this map and the values obtained by applying the
[transform]\n * function to each entry in this [Map].\n *\n * The returned map preserves the entry iteration order of
the original map.\n *\n * @sample samples.collections.Maps.Transformations.mapValues\n *\npublic inline fun
<K, V, R> Map<out K, V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {\n return
mapValuesTo(LinkedHashMap<K, R>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n}\n\n/**\n * Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this\n * [Map] and the values of this map.\n *\n * In case if any two entries are mapped to the equal keys, the value of the
latter one will overwrite\n * the value associated with the former one.\n *\n * The returned map
preserves the entry iteration order of the original map.\n *\n * @sample
samples.collections.Maps.Transformations.mapKeys\n *\npublic inline fun <K, V, R> Map<out K,
V>.mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {\n return mapKeysTo(LinkedHashMap<R,
V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n}\n\n/**\n * Returns a map containing all key-
value pairs with keys matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of
the original map.\n *\n * @sample samples.collections.Maps.Filtering.filterKeys\n *\npublic inline fun <K, V>
Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n
for (entry in this) {\n if (predicate(entry.key)) {\n result.put(entry.key, entry.value)\n }\n }\n
return result\n}\n}\n\n/**\n * Returns a map containing all key-value pairs with values matching the given
[predicate].\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n * @sample samples.collections.Maps.Filtering.filterValues\n *\npublic inline
fun <K, V> Map<out K, V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {\n val result =
LinkedHashMap<K, V>()\n for (entry in this) {\n if (predicate(entry.value)) {\n result.put(entry.key,
entry.value)\n }\n }\n return result\n}\n}\n\n/**\n * Appends all entries matching the given [predicate] into
the mutable map given as [destination] parameter.\n *\n * @return the destination map.\n *\n * @sample
samples.collections.Maps.Filtering.filterTo\n *\npublic inline fun <K, V, M : MutableMap<in K, in V>> Map<out
K, V>.filterTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {\n for (element in this) {\n if
(predicate(element)) {\n destination.put(element.key, element.value)\n }\n }\n return
destination\n}\n}\n\n/**\n * Returns a new map containing all key-value pairs matching the given [predicate].\n
*\n * The returned map preserves the entry iteration order of the original map.\n *\n * @sample
samples.collections.Maps.Filtering.filter\n *\npublic inline fun <K, V> Map<out K, V>.filter(predicate:
(Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return filterTo(LinkedHashMap<K, V>(),
predicate)\n}\n}\n\n/**\n * Appends all entries not matching the given [predicate] into the given [destination].\n *\n *
@return the destination map.\n *\n * @sample samples.collections.Maps.Filtering.filterNotTo\n *\npublic inline fun
<K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterNotTo(destination: M, predicate: (Map.Entry<K, V>) -
> Boolean): M {\n for (element in this) {\n if (!predicate(element)) {\n destination.put(element.key,
element.value)\n }\n }\n return destination\n}\n}\n\n/**\n * Returns a new map containing all key-value pairs
not matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original
map.\n *\n * @sample
samples.collections.Maps.Filtering.filterNot\n *\npublic inline fun <K, V> Map<out K, V>.filterNot(predicate:
(Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return filterNotTo(LinkedHashMap<K, V>(),
predicate)\n}\n}\n\n/**\n * Returns a new map containing all key-value pairs from the given collection of pairs.\n *\n *
The returned map preserves the entry iteration order of the original collection.\n *\n * If any of two pairs would have the
same key the last one gets added to the map.\n *\npublic fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V>
{\n if (this is Collection) {\n return when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(if (this is

```

```

List) this[0] else iterator().next())\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n }\n}\n return toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given collection of pairs.\n */\npublic fun\n<K, V, M : MutableMap<in K, in V>> Iterable<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n}\n\n/**\n * Returns a new map containing all key-value pairs from the given array of pairs.\n */\n * The returned map preserves the entry iteration order of the original array.\n */\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> =\n when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(this[0])\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n }\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given array of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Array<out Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n}\n\n/**\n * Returns a new map containing all key-value pairs from the given sequence of pairs.\n */\n * The returned map preserves the entry iteration order of the original sequence.\n */\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> =\n toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given sequence of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Sequence<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n}\n\n/**\n * Returns a new read-only map containing all key-value pairs from the original map.\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMap(): Map<K, V> =\n when (size) {\n 0 -> emptyMap()\n 1 -> toSingletonMap()\n else -> toMutableMap()\n }\n\n/**\n * Returns a new mutable map containing all key-value pairs from the original map.\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMutableMap(): MutableMap<K, V> =\n LinkedHashMap(this)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n}\n\n/**\n * Creates a new read-only map by replacing or adding an entry to this map from a given key-value [pair].\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n * The [pair] is iterated in the end if it has a unique key.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pair: Pair<K, V>): Map<K, V> =\n if (this.isEmpty()) mapOf(pair) else\n LinkedHashMap(this).apply { put(pair.first, pair.second) }\n}\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given collection of key-value [pairs].\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Iterable<Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else\n LinkedHashMap(this).apply { putAll(pairs) }\n}\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given array of key-value [pairs].\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] array.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Array<out Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else\n LinkedHashMap(this).apply { putAll(pairs) }\n}\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given sequence of key-value [pairs].\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] sequence.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Sequence<Pair<K, V>>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(pairs) }.optimizeReadOnlyMap()\n}\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from another [map].\n */\n * The returned map preserves the entry iteration order of the original map.\n */\n * Those entries of another [map] that are missing in this map are iterated in the end in the order of that [map].\n */\npublic operator fun <K, V> Map<out K, V>.plus(map: Map<out K, V>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(map) }\n}\n\n/**\n * Appends or replaces the given [pair] in this mutable

```

```

map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in
V>.plusAssign(pair: Pair<K, V>) {\n put(pair.first, pair.second)\n}\n\n/**\n * Appends or replaces all pairs from
the given collection of
[pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in
K, in V>.plusAssign(pairs: Iterable<Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs
from the given array of [pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun
<K, V> MutableMap<in K, in V>.plusAssign(pairs: Array<out Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n *
Appends or replaces all pairs from the given sequence of [pairs] in this mutable map.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs:
Sequence<Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or replaces all entries from the given [map] in
this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in
V>.plusAssign(map: Map<K, V>) {\n putAll(map)\n}\n\n/**\n * Returns a map containing all entries of the
original map except the entry
with the given [key].\n *\n * The returned map preserves the entry iteration order of the original map.\n
*\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K, V>.minus(key: K): Map<K, V> =\n
this.toMutableMap().apply { minusAssign(key) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all
entries of the original map except those entries\n * the keys of which are contained in the given [keys] collection.\n
*\n * The returned map preserves the entry iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic
operator fun <K, V> Map<out K, V>.minus(keys: Iterable<K>): Map<K, V> =\n this.toMutableMap().apply {
minusAssign(keys) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map
except those entries\n * the keys of which are contained in the given [keys] array.\n *\n * The returned map
preserves the entry iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V>
Map<out K, V>.minus(keys:
Array<out K>): Map<K, V> =\n this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] sequence.\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(keys: Sequence<K>): Map<K, V> =\n this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Removes the entry with the given [key] from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(key: K) {\n remove(key)\n}\n\n/**\n * Removes all entries the keys of which are contained in
the given [keys] collection from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys:
Iterable<K>) {\n this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries the keys of which are contained in
the given [keys] array from this mutable map.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<K, V>.minusAssign(keys: Array<out K>) {\n
this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries from the keys of which are contained in the given
[keys] sequence from this mutable map.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline
operator fun <K, V> MutableMap<K, V>.minusAssign(keys: Sequence<K>) {\n
this.keys.removeAll(keys)\n}\n\n// do not expose for now @PublishedApi\ninternal fun <K, V> Map<K,
V>.optimizeReadOnlyMap() = when (size) {\n 0 -> emptyMap()\n 1 -> toSingletonMapOrSelf()\n else ->
this\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found
in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n@file:OptIn(kotlin.experimenta
l.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\ninternal object
EmptySet : Set<Nothing>, Serializable {\n private const val serialVersionUID: Long =

```

```

3406603774387020532\n\n override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()\n
override fun hashCode(): Int = 0\n override fun toString(): String = \"[]\"\n\n override val size: Int get() = 0\n
override fun isEmpty(): Boolean = true\n override fun contains(element: Nothing): Boolean = false\n override
fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()\n\n override fun iterator():
Iterator<Nothing> = EmptyIterator\n\n private fun readResolve(): Any = EmptySet\n}\n\n/**\n * Returns an
empty read-only set. The returned set is serializable (JVM).\n * @sample
samples.collections.Collections.Sets.emptyReadOnlySet\n
*/\n\npublic fun <T> emptySet(): Set<T> = EmptySet\n\n/**\n * Returns a new read-only set with the given
elements.\n * Elements of the set are iterated in the order they were specified.\n * The returned set is serializable
(JVM).\n * @sample samples.collections.Collections.Sets.readOnlySet\n */\n\npublic fun <T> setOf(vararg elements:
T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()\n\n/**\n * Returns an empty read-only set. The
returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> setOf(): Set<T> = emptySet()\n\n/**\n * Returns an empty
new [MutableSet].\n * The returned set preserves the element iteration order.\n * @sample
samples.collections.Collections.Sets.emptyMutableSet\n
*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableSetOf(): MutableSet<T> =
LinkedHashSet()\n\n/**\n *
Returns a new [MutableSet] with the given elements.\n * Elements of the set are iterated in the order they were
specified.\n * @sample samples.collections.Collections.Sets.mutableSet\n */\n\npublic fun <T> mutableSetOf(vararg
elements: T): MutableSet<T> = elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/** Returns
an empty new [HashSet].\n * @SinceKotlin("1.1")\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>
hashSetOf(): HashSet<T> = HashSet()\n\n/** Returns a new [HashSet] with the given elements.\n */\n\npublic fun <T>
hashSetOf(vararg elements: T): HashSet<T> =
elements.toCollection(HashSet(mapCapacity(elements.size)))\n\n/**\n * Returns an empty new [LinkedHashSet].\n
*/\n * @sample samples.collections.Collections.Sets.emptyLinkedHashSet\n
*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> linkedSetOf(): LinkedHashSet<T>
= LinkedHashSet()\n\n/**\n * Returns a new [LinkedHashSet] with the given elements.\n * Elements of the set are
iterated
in the order they were specified.\n * @sample samples.collections.Collections.Sets.linkedHashSet\n */\n\npublic fun
<T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/**\n * Returns a new read-only set either
with single given element, if it is not null, or empty set if the element is null.\n * The returned set is serializable
(JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n */\n\n@SinceKotlin("1.4")\n\npublic fun <T
: Any> setOfNotNull(element: T?): Set<T> = if (element != null) setOf(element) else emptySet()\n\n/**\n * Returns
a new read-only set only with those given elements, that are not null.\n * Elements of the set are iterated in the order
they were specified.\n * The returned set is serializable (JVM).\n * @sample
samples.collections.Collections.Sets.setOfNotNull\n */\n\n@SinceKotlin("1.4")\n\npublic fun <T : Any>
setOfNotNull(vararg elements: T?): Set<T> {\n return elements.filterNotNullTo(LinkedHashSet())\n}\n\n/**\n * Builds a
new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a
read-only set with the same elements.\n * The set passed as a receiver to the [builderAction] is valid only inside
that function.\n * Using it outside of the function produces an unspecified behavior.\n * Elements of the set are
iterated in the order they were added by the [builderAction].\n * The returned set is serializable (JVM).\n *
*/\n * @sample samples.collections.Builders.Sets.buildSetSample\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\n\npublic inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>().()
> Unit): Set<E> {\n contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n return
buildSetInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n\ninternal
expect

```

```

inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {
 * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction] and returning a read-only set with the same elements.
 * The set passed as a receiver to the [builderAction] is valid only inside that function.
 * Using it outside of the function produces an unspecified behavior.
 * [capacity] is used to hint the expected number of elements added in the [builderAction].
 * Elements of the set are iterated in the order they were added by the [builderAction].
 * The returned set is serializable (JVM).
 * @throws IllegalArgumentException if the given [capacity] is negative.
 * @sample samples.collections.Builders.Sets.buildSetSample
}

* Since Kotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION")
public inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {
 contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }
 return buildSetInternal(capacity, builderAction)
}

* Since Kotlin("1.3")
@kotlin.internal.InlineOnly
internal expect inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit): Set<E> {
 Returns this Set if it's not `null` and the empty set otherwise.
}

* Since Kotlin("1.7")
@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.optimizeReadOnlySet() = when (size) {
 0 -> emptySet()
 1 -> setOf(iterator().next())
 else -> this
}

/* Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/
// Auto-generated file. DO NOT EDIT!
package kotlin.ranges

* A range of values of type `Char`.

* Since Kotlin("1.7")
@ExperimentalStdlibApi
public class CharRange(start: Char, endInclusive: Char) : CharProgression(start, endInclusive, 1), ClosedRange<Char>, OpenEndRange<Char> {
 override val start: Char get() = first
 override val endInclusive: Char get() = last
 @Deprecated("Can throw an exception when it's impossible to represent the value with Char type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.")
 @SinceKotlin("1.7")
 @ExperimentalStdlibApi
 override val endExclusive: Char get() {
 if (last == Char.MAX_VALUE)
 error("Cannot return the exclusive upper bound of a range that includes MAX_VALUE.")
 return last + 1
 }
 override fun contains(value: Char): Boolean = first <= value && value <= last
 /** Checks whether the range is empty.
 * The range is empty if its start value is greater than the end value.
 */
 override fun isEmpty(): Boolean = first > last
 override fun equals(other: Any?): Boolean =
 other is CharRange && (isEmpty() && other.isEmpty() || first == other.first && last == other.last)
 override fun hashCode(): Int =
 if (isEmpty()) -1 else (31 * first.code + last.code)
 override fun toString(): String = "$first..$last"
 companion object {
 /** An empty range of values of type Char.
 */
 public val EMPTY: CharRange = CharRange(1.toChar(), 0.toChar())
 }
}

* A range of values of type `Int`.

* Since Kotlin("1.7")
@ExperimentalStdlibApi
public class IntRange(start: Int, endInclusive: Int) : IntProgression(start, endInclusive, 1), ClosedRange<Int>, OpenEndRange<Int> {
 override val start: Int get() = first
 override val endInclusive: Int get() = last
 @Deprecated("Can throw an exception when it's impossible to represent the value with Int type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.")
 @SinceKotlin("1.7")
 @ExperimentalStdlibApi
 override val endExclusive: Int get() {
 if (last == Int.MAX_VALUE)
 error("Cannot return the exclusive upper bound of a range that includes MAX_VALUE.")
 return last + 1
 }
 override fun contains(value: Int): Boolean = first <= value && value <= last
 /** Checks whether the range is empty.
 * The range is empty if its start value is greater than the end value.
 */
 override fun isEmpty(): Boolean = first > last
 override fun equals(other: Any?): Boolean =
 other is IntRange && (isEmpty() && other.isEmpty() || first == other.first && last == other.last)
 override fun hashCode(): Int =
 if (isEmpty()) -1 else (31 * first + last)
 override fun toString(): String = "$first..$last"
 companion object {
 /** An empty range of values of type

```

```

Int. */n public val EMPTY: IntRange = IntRange(1, 0)\n }\n}\n\n/**\n * A range of values of type `Long`.\n */\n@OptIn(ExperimentalStdlibApi::class)\npublic class LongRange(start: Long, endInclusive: Long) :
LongProgression(start, endInclusive, 1), ClosedRange<Long>, OpenEndRange<Long> {\n override val start:
Long get() = first\n override val endInclusive: Long get() = last\n \n @Deprecated("Can throw an exception
when it's impossible to represent the value with Long type, for example, when the range includes MAX_VALUE.
It's recommended to use 'endInclusive' property that doesn't throw.")\n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n override val endExclusive: Long get() {\n if (last == Long.MAX_VALUE)
error("Cannot return the exclusive upper bound of a range that includes MAX_VALUE.")\n return last + 1\n }\n \n override fun contains(value: Long): Boolean = first <= value && value <= last\n \n /**\n * Checks
whether the
range is empty.\n *\n * The range is empty if its start value is greater than the end value.\n */\n override
fun isEmpty(): Boolean = first > last\n \n override fun equals(other: Any?): Boolean =\n other is LongRange
&& (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last\n \n override fun hashCode():
Int =\n if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))).toInt()\n \n override fun
toString(): String = "$first..$last"\n \n companion object {\n /** An empty range of values of type Long. */\n public val EMPTY: LongRange = LongRange(1, 0)\n }\n}\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n@file:Suppress("PLATFORM_CLASS_MAPPED_TO_KOTLIN")\n\npackage
kotlin.text\n\n/**\n * Parses the string as a signed [Byte] number and returns the result\n * or `null` if the string is
not a valid representation of a number.\n */\n@SinceKotlin("1.1")\npublic fun String.toByteArrayOrNull(): Byte? =
toByteOrNull(radix = 10)\n\n/**\n * Parses the string as a signed [Byte] number and returns the result\n * or `null`
if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is
not a valid radix for string to number conversion.\n */\n@SinceKotlin("1.1")\npublic fun
String.toByteArrayOrNull(radix: Int): Byte? {\n val int = this.toIntOrNull(radix) ?: return null\n if (int <
Byte.MIN_VALUE || int > Byte.MAX_VALUE) return null\n return int.toByte()\n}\n\n/**\n * Parses the string as
a [Short] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*/\n@SinceKotlin("1.1")\npublic fun String.toShortOrNull(): Short? = toShortOrNull(radix
= 10)\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the string is not a valid
representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string
to number conversion.\n */\n@SinceKotlin("1.1")\npublic fun String.toShortOrNull(radix: Int): Short? {\n val int
= this.toIntOrNull(radix) ?: return null\n if (int < Short.MIN_VALUE || int > Short.MAX_VALUE) return null\n
return int.toShort()\n}\n\n/**\n * Parses the string as an [Int] number and returns the result\n * or `null` if the string
is not a valid representation of a number.\n */\n@SinceKotlin("1.1")\npublic fun String.toIntOrNull(): Int? =
toIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [Int] number and returns the result\n * or `null` if the
string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a
valid radix for string to number conversion.\n */\n@SinceKotlin("1.1")\npublic
fun String.toIntOrNull(radix: Int): Int? {\n checkRadix(radix)\n \n val length = this.length\n if (length == 0)
return null\n \n val start: Int\n val isNegative: Boolean\n val limit: Int\n val firstChar = this[0]\n if
(firstChar < '0') { // Possible leading sign\n if (length == 1) return null // non-digit (possible sign) only, no
digits after\n start = 1\n if (firstChar == '-') {\n isNegative = true\n limit =
Int.MIN_VALUE\n } else if (firstChar == '+') {\n isNegative = false\n limit = -
Int.MAX_VALUE\n } else\n return null\n } else {\n start = 0\n isNegative = false\n limit
= -Int.MAX_VALUE\n }\n \n val limitForMaxRadix = (-Int.MAX_VALUE) / 36\n var limitBeforeMul =
limitForMaxRadix\n var result = 0\n for (i in start until length) {\n val digit = digitOf(this[i], radix)\n if (digit

```





```

duration whose value is positive infinity. It is useful for representing timeouts that should never expire. */
public val INFINITE: Duration = durationOfMillis(MAX_MILLIS) internal val NEG_INFINITE: Duration =
durationOfMillis(-MAX_MILLIS) /** Converts the given time duration [value] expressed in the specified
[sourceUnit] into the specified [targetUnit]. */ @ExperimentalTime public fun convert(value: Double,
sourceUnit: DurationUnit, targetUnit: DurationUnit): Double = convertDurationUnit(value, sourceUnit,
targetUnit) // Duration construction extension properties in Duration companion scope /** Returns a
[Duration] equal to this [Int] number of nanoseconds.
*/ @kotlin.internal.InlineOnly public inline val Int.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS) /** Returns a [Duration] equal to this [Long] number of
nanoseconds. */ @kotlin.internal.InlineOnly public inline val Long.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS) /**
* Returns a [Duration] equal to this [Double]
number of nanoseconds.
* Depending on its magnitude, the value is rounded to an integer number of
nanoseconds or milliseconds.
* @throws IllegalArgumentException if this [Double] value is
`NaN`.
*/ @kotlin.internal.InlineOnly public inline val Double.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS) /** Returns a [Duration] equal to this [Int] number of
microseconds. */ @kotlin.internal.InlineOnly public inline val Int.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)
/** Returns a [Duration] equal to this [Long] number of microseconds. */ @kotlin.internal.InlineOnly
public inline val Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS) /**
* Returns a [Duration] equal to this [Double] number of microseconds.
* Depending on its magnitude,
the value is rounded to an integer number of nanoseconds or milliseconds.
* @throws
IllegalArgumentException if this [Double] value is `NaN`.
*/ @kotlin.internal.InlineOnly public
inline val Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS) /** Returns a
[Duration] equal to this [Int] number of milliseconds. */ @kotlin.internal.InlineOnly public inline val
Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS) /** Returns a [Duration] equal to this
[Long] number of milliseconds. */ @kotlin.internal.InlineOnly
public inline val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS) /**
* Returns a [Duration] equal to this [Double] number of milliseconds.
* Depending on its magnitude,
the value is rounded to an integer number of nanoseconds or milliseconds.
* @throws
IllegalArgumentException if this [Double] value is `NaN`.
*/ @kotlin.internal.InlineOnly public
inline val Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS) /** Returns a
[Duration] equal to this [Int] number of seconds. */ @kotlin.internal.InlineOnly public inline val
Int.seconds get() = toDuration(DurationUnit.SECONDS) /** Returns a [Duration] equal to this [Long]
number of seconds. */ @kotlin.internal.InlineOnly public inline val Long.seconds get() =
toDuration(DurationUnit.SECONDS) /**
* Returns a [Duration] equal
to this [Double] number of seconds.
* Depending on its magnitude, the value is rounded to an
integer number of nanoseconds or milliseconds.
* @throws IllegalArgumentException if this
[Double] value is `NaN`.
*/ @kotlin.internal.InlineOnly public inline val Double.seconds get() =
toDuration(DurationUnit.SECONDS) /** Returns a [Duration] equal to this [Int] number of minutes. */
@kotlin.internal.InlineOnly public inline val Int.minutes get() = toDuration(DurationUnit.MINUTES)
/** Returns a [Duration] equal to this [Long] number of minutes. */ @kotlin.internal.InlineOnly
public inline val Long.minutes get() = toDuration(DurationUnit.MINUTES) /**
* Returns a
[Duration] equal to this [Double] number of minutes.
* Depending on its magnitude, the value is
rounded to an integer number of nanoseconds or milliseconds.
*
* @throws IllegalArgumentException if this [Double] value is `NaN`.
*/
@kotlin.internal.InlineOnly public inline val Double.minutes get() =
toDuration(DurationUnit.MINUTES) /** Returns a [Duration] equal to this [Int] number of hours. */
@kotlin.internal.InlineOnly public inline val Int.hours get() = toDuration(DurationUnit.HOURS) /**

```

```

Returns a [Duration] equal to this [Long] number of hours. */\n @kotlin.internal.InlineOnly\n public inline
val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n /**\n * Returns a [Duration] equal to this
[Double] number of hours.\n *\n * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.hours get()
= toDuration(DurationUnit.HOURS)\n\n /** Returns a [Duration] equal to this [Int] number of days. */\n
 @kotlin.internal.InlineOnly\n public inline val Int.days get() = toDuration(DurationUnit.DAYS)\n\n /**
Returns a [Duration] equal to this [Long] number of days. */\n @kotlin.internal.InlineOnly\n public inline
val Long.days get() = toDuration(DurationUnit.DAYS)\n\n /**\n * Returns a [Duration] equal to this
[Double] number of days.\n *\n * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.days get() =
toDuration(DurationUnit.DAYS)\n\n // deprecated static factory functions\n\n /** Returns a [Duration]
representing the specified [value] number of nanoseconds. */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use 'Int.nanoseconds' extension property from
Duration.Companion instead.", ReplaceWith("value.nanoseconds"),
"\kotlin.time.Duration.Companion.nanoseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6",
errorSince = "1.8")\n public fun nanoseconds(value: Int): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n /** Returns a [Duration] representing the specified
[value] number of nanoseconds. */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use 'Long.nanoseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.nanoseconds", "\kotlin.time.Duration.Companion.nanoseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun nanoseconds(value: Long):
Duration = value.toDuration(DurationUnit.NANOSECONDS)\n\n /**\n * Returns a [Duration]
representing the specified
[value] number of nanoseconds.\n *\n * @throws IllegalArgumentException if the provided `Double`
[value] is `NaN`.\n */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use
'Double.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("value.nanoseconds",
"\kotlin.time.Duration.Companion.nanoseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6",
errorSince = "1.8")\n public fun nanoseconds(value: Double): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n /** Returns a [Duration] representing the specified
[value] number of microseconds. */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use 'Int.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "\kotlin.time.Duration.Companion.microseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun microseconds(value: Int): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n /**
Returns a [Duration] representing the specified [value] number of microseconds. */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use 'Long.microseconds' extension property from
Duration.Companion instead.", ReplaceWith("value.microseconds",
"\kotlin.time.Duration.Companion.microseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6",
errorSince = "1.8")\n public fun microseconds(value: Long): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n /**\n * Returns a [Duration] representing the
specified [value] number of microseconds.\n *\n * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n */\n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use 'Double.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "\kotlin.time.Duration.Companion.microseconds"))\n @DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun microseconds(value:
Double): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n /** Returns a [Duration]

```

```

representing the specified [value] number of milliseconds. */n @SinceKotlin("1.5")\n
@ExperimentalTime\n @Deprecated("Use 'Int.milliseconds' extension property from Duration.Companion
instead.", ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun milliseconds(value: Int):
Duration = value.toDuration(DurationUnit.MILLISECONDS)\n\n /** Returns a [Duration] representing the
specified [value] number of milliseconds. */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Long.milliseconds' extension
property from Duration.Companion instead.", ReplaceWith("value.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))n @DeprecatedSinceKotlin(warningSince = "1.6",
errorSince = "1.8")\n public fun milliseconds(value: Long): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n /**\n * Returns a [Duration] representing the
specified [value] number of milliseconds.\n */n * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Double.milliseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun milliseconds(value:
Double): Duration = value.toDuration(DurationUnit.MILLISECONDS)\n\n /** Returns a [Duration]
representing the specified [value] number of seconds. */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Int.seconds' extension property from Duration.Companion instead.",
ReplaceWith("value.seconds", "kotlin.time.Duration.Companion.seconds"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun seconds(value: Int):
Duration = value.toDuration(DurationUnit.SECONDS)\n\n /** Returns a [Duration] representing the specified
[value] number of seconds. */n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use
'Long.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"kotlin.time.Duration.Companion.seconds"))n @DeprecatedSinceKotlin(warningSince = "1.6", errorSince =
"1.8")\n public fun seconds(value: Long): Duration = value.toDuration(DurationUnit.SECONDS)\n\n
/**\n * Returns a [Duration]
representing the specified [value] number of seconds.\n */n * @throws IllegalArgumentException if the
provided `Double` [value] is `NaN`.\n */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Double.seconds' extension property from Duration.Companion instead.",
ReplaceWith("value.seconds", "kotlin.time.Duration.Companion.seconds"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun seconds(value: Double):
Duration = value.toDuration(DurationUnit.SECONDS)\n\n\n /** Returns a [Duration] representing the
specified [value] number of minutes. */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Int.minutes' extension property from Duration.Companion instead.",
ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun minutes(value:
Int): Duration = value.toDuration(DurationUnit.MINUTES)\n\n /** Returns a [Duration] representing the
specified [value] number of minutes. */n @SinceKotlin("1.5")\n @ExperimentalTime\n
@Deprecated("Use 'Long.minutes' extension property from Duration.Companion instead.",
ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))n
@DeprecatedSinceKotlin(warningSince = "1.6", errorSince = "1.8")\n public fun minutes(value: Long):
Duration = value.toDuration(DurationUnit.MINUTES)\n\n /**\n * Returns a [Duration] representing the
specified [value] number of minutes.\n */n * @throws IllegalArgumentException if the provided `Double`
[value] is `NaN`.\n */n @SinceKotlin("1.5")\n @ExperimentalTime\n @Deprecated("Use
'Double.minutes' extension property from Duration.Companion instead.", ReplaceWith("value.minutes",
"kotlin.time.Duration.Companion.minutes"))n

```

```

@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun minutes(value:
Double): Duration = value.toDuration(DurationUnit.MINUTES)\n\n /** Returns a [Duration] representing the
specified [value] number of hours. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Int.hours' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.hours\", \"kotlin.time.Duration.Companion.hours\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun hours(value: Int): Duration
= value.toDuration(DurationUnit.HOURS)\n\n /** Returns a [Duration] representing the specified [value]
number of hours. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use
'Long.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"value.hours\",
\"kotlin.time.Duration.Companion.hours\"))\n @DeprecatedSinceKotlin(warningSince
= \"1.6\", errorSince = \"1.8\")\n public fun hours(value: Long): Duration =
value.toDuration(DurationUnit.HOURS)\n\n /**\n * Returns a [Duration] representing the specified
[value] number of hours.\n * \n * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Double.hours'
extension property from Duration.Companion instead.\", ReplaceWith(\"value.hours\",
\"kotlin.time.Duration.Companion.hours\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince =
\"1.8\")\n public fun hours(value: Double): Duration = value.toDuration(DurationUnit.HOURS)\n\n /**
Returns a [Duration] representing the specified [value] number of days. */\n @SinceKotlin(\"1.5\")\n
@ExperimentalTime\n @Deprecated(\"Use 'Int.days' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.days\",
\"kotlin.time.Duration.Companion.days\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince =
\"1.8\")\n public fun days(value: Int): Duration = value.toDuration(DurationUnit.DAYS)\n\n /** Returns a
[Duration] representing the specified [value] number of days. */\n @SinceKotlin(\"1.5\")\n
@ExperimentalTime\n @Deprecated(\"Use 'Long.days' extension property from Duration.Companion
instead.\", ReplaceWith(\"value.days\", \"kotlin.time.Duration.Companion.days\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun days(value: Long):
Duration = value.toDuration(DurationUnit.DAYS)\n\n /**\n * Returns a [Duration] representing the
specified [value] number of days.\n * \n * @throws IllegalArgumentException if the provided `Double`
[value] is `NaN`.\n */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use
'Double.days' extension property from Duration.Companion instead.\", ReplaceWith(\"value.days\",
\"kotlin.time.Duration.Companion.days\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince =
\"1.8\")\n public fun days(value: Double): Duration = value.toDuration(DurationUnit.DAYS)\n\n /**\n * Parses a string that represents a duration and returns the parsed [Duration] value.\n * \n * The following
formats are accepted:\n * \n * - ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and
[parseIsoString].\n * \n * - The format of string returned by the default [Duration.toString] and `toString` in a
specific unit,\n * \n * e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n * \n * @throws IllegalArgumentException if
the string doesn't represent a duration in any of the supported formats.\n * \n * @sample
samples.time.Durations.parse\n */\n public fun parse(value: String): Duration
= try {\n parseDuration(value, strictIso = false)\n } catch (e: IllegalArgumentException) {\n
throw IllegalArgumentException(\"Invalid duration string format: '$value'.\", e)\n }\n\n /**\n * Parses
a string that represents a duration in a restricted ISO-8601 composite representation\n * and returns the parsed
[Duration] value.\n * \n * Composite representation is a relaxed version of ISO-8601 duration format that
supports\n * \n * negative durations and negative values of individual components.\n * \n * The following
restrictions are imposed:\n * \n * - The only allowed non-time designator is days (`D`), `Y` (years), `W`
(weeks), and `M` (months) are not supported.\n * \n * - Day is considered to be exactly 24 hours (24-hour clock
time scale).\n * \n * - Alternative week-based representation `[\"P\"][number][\"W\"]` is not supported.\n * \n
 * @throws IllegalArgumentException

```

```

if the string doesn't represent a duration in ISO-8601 format.\n * @sample
samples.time.Durations.parseIsoString\n */\n public fun parseIsoString(value: String): Duration = try {\n
 parseDuration(value, strictIso = true)\n } catch (e: IllegalArgumentException) {\n throw
IllegalArgumentExcep\n tion(\n "Invalid ISO duration string format: '$value'.", e)\n }\n /**\n * Parses a
string that represents a duration and returns the parsed [Duration] value,\n * or `null` if the string doesn't
represent a duration in any of the supported formats.\n *\n * The following formats are accepted:\n
*\n * - Restricted ISO-8601 duration composite representation, e.g. `P1DT2H3M4.058S`, see [toIsoString] and
[parseIsoString].\n * - The format of string returned by the default [Duration.toString] and `toString` in a
specific unit,\n * e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n
*\n * @sample samples.time.Durations.parse\n */\n public fun parseOrNull(value: String): Duration? =
try {\n parseDuration(value, strictIso = false)\n } catch (e: IllegalArgumentException) {\n null\n
 }\n /**\n * Parses a string that represents a duration in restricted ISO-8601 composite representation\n
*\n * and returns the parsed [Duration] value or `null` if the string doesn't represent a duration in the format\n
*\n * acceptable by [parseIsoString].\n *\n * @sample samples.time.Durations.parseIsoString\n */\n
public fun parseIsoStringOrNull(value: String): Duration? = try {\n parseDuration(value, strictIso = true)\n
 } catch (e: IllegalArgumentException) {\n null\n }\n // arithmetic operators\n /** Returns
the negative of this value. *\n public operator fun unaryMinus(): Duration = durationOf(-value,
unitDiscriminator)\n *\n /**\n * Returns a duration whose value is the sum of this and [other] duration values.\n
*\n * @throws
IllegalArgumentExcep\n tion if the operation results in an undefined value for the given arguments,\n * e.g. when
adding infinite durations of different sign.\n *\n public operator fun plus(other: Duration): Duration {\n
 when {\n this.isInfinite() -> {\n if (other.isFinite() || (this.rawValue xor other.rawValue >= 0))\n
 return this\n else\n throw IllegalArgumentExcep\n tion(\n
 "Summing infinite durations of\n different signs yields an undefined result.")\n
 }\n other.isInfinite() -> return other\n }\n return when {\n this.unitDiscriminator == other.unitDiscriminator -> {\n val result = this.value +
other.value // never overflows long, but can overflow long63\n when {\n isInNanos()\n
 ->\n durationOfNanosNormalized(result)\n else ->\n
 durationOfMillisNormalized(result)\n }\n }\n this.isInMillis() ->\n
addValuesMixedRanges(this.value, other.value)\n else ->\n addValuesMixedRanges(other.value,
this.value)\n }\n }\n private fun addValuesMixedRanges(thisMillis: Long, otherNanos: Long): Duration
{\n val otherMillis = nanosToMillis(otherNanos)\n val resultMillis = thisMillis + otherMillis\n return if
(resultMillis in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n val otherNanoRemainder =
otherNanos - millisToNanos(otherMillis)\n durationOfNanos(millisToNanos(resultMillis) +
otherNanoRemainder)\n } else {\n durationOfMillis(resultMillis.coerceIn(-MAX_MILLIS,
MAX_MILLIS))\n }\n }\n /**\n * Returns a duration whose value is the difference
between this and [other] duration values.\n *\n * @throws IllegalArgumentExcep\n tion if the operation results in an undefined value for the given arguments,\n * e.g. when subtracting infinite durations of the same sign.\n *\n public operator fun minus(other: Duration): Duration = this + (-other)\n /**\n * Returns a duration whose
value is this duration value multiplied by the given [scale] number.\n *\n * @throws IllegalArgumentExcep\n tion if the operation results in an undefined value for the given arguments,\n * e.g. when multiplying an infinite
duration by zero.\n *\n public operator fun times(scale: Int): Duration {\n if (isInfinite()) {\n return
when {\n scale == 0 -> throw IllegalArgumentExcep\n tion(\n "Multiplying infinite duration by zero yields an\n
 undefined result.")\n scale > 0 -> this\n else -> -this\n }\n }\n if (scale == 0)\n return ZERO\n val value = value\n val result = value * scale\n return if (isInNanos()) {\n if (value in
(MAX_NANOS / Int.MIN_VALUE)..(-MAX_NANOS / Int.MIN_VALUE)) {\n // can't overflow nanos
range for any scale\n durationOfNanos(result)\n } else {\n if (result / scale == value) {\n
 durationOfNanosNormalized(result)\n } else {\n val millis = nanosToMillis(value)\n

```

```

 val remNanos = value - millisToNanos(millis)\n val resultMillis = millis * scale\n val totalMillis = resultMillis + nanosToMillis(remNanos * scale)\n if (resultMillis / scale == millis && totalMillis xor resultMillis >= 0) {\n durationOfMillis(totalMillis.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n } else {\n if (value.sign * scale.sign > 0) INFINITE else NEG_INFINITE\n }\n }\n }\n } else {\n if (result / scale == value) {\n durationOfMillis(result.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n } else {\n if (value.sign * scale.sign > 0) INFINITE else NEG_INFINITE\n }\n }\n}\n\n/**\n * Returns a duration whose value is this duration value multiplied by the given [scale] number.\n * The operation may involve rounding when the result cannot be represented exactly with a [Double] number.\n * @throws\n * IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when multiplying an infinite duration by zero.\n */\n public operator fun times(scale: Double): Duration {\n val intScale = scale.roundToInt()\n if (intScale.toDouble() == scale) {\n return times(intScale)\n }\n val unit = storageUnit\n val result = toDouble(unit) * scale\n return result.toDuration(unit)\n }\n\n/**\n * Returns a duration whose value is this duration value divided by the given [scale] number.\n * @throws\n * IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when dividing zero duration by zero.\n */\n public operator fun div(scale: Int): Duration {\n if (scale == 0) {\n return when {\n isPositive() -> INFINITE\n isNegative() -> NEG_INFINITE\n else -> throw\n IllegalArgumentException("Dividing zero duration by zero yields an undefined result.")\n }\n }\n if (isInfinite()) {\n return this * scale.sign\n }\n val result = value / scale\n if (result in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n val rem = millisToNanos(value - (result * scale)) / scale\n return durationOfNanos(millisToNanos(result) + rem)\n }\n return durationOfMillis(result)\n }\n\n/**\n * Returns a duration whose value is this duration value divided by the given [scale] number.\n * @throws\n * IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when dividing an infinite duration by infinity or zero duration by zero.\n */\n public operator fun div(scale: Double): Duration {\n val intScale = scale.roundToInt()\n if (intScale.toDouble() == scale && intScale != 0) {\n return div(intScale)\n }\n val unit = storageUnit\n val result = toDouble(unit) / scale\n return result.toDuration(unit)\n }\n\n/**\n * Returns a number that is the ratio of this and [other] duration values.\n */\n public operator fun div(other: Duration): Double {\n {\n val coarserUnit = maxOf(this.storageUnit, other.storageUnit)\n return this.toDouble(coarserUnit) / other.toDouble(coarserUnit)\n }\n\n /**\n * Returns true, if the duration value is less than zero.\n */\n public fun isNegative(): Boolean = rawValue < 0\n\n /**\n * Returns true, if the duration value is greater than zero.\n */\n public fun isPositive(): Boolean = rawValue > 0\n\n /**\n * Returns true, if the duration value is infinite.\n */\n public fun isInfinite(): Boolean = rawValue == INFINITE.rawValue || rawValue == NEG_INFINITE.rawValue\n\n /**\n * Returns true, if the duration value is finite.\n */\n public fun isFinite(): Boolean = !isInfinite()\n\n /**\n * Returns the absolute value of this value. The returned value is always non-negative.\n */\n public val absoluteValue: Duration\n\n get() = if (isNegative()) -this else this\n\n override fun compareTo(other: Duration): Int {\n val compareBits = this.rawValue xor other.rawValue\n if (compareBits < 0 || compareBits.toInt() and 1 == 0) // different signs or same sign/same range\n return this.rawValue.compareTo(other.rawValue)\n // same sign/different ranges\n val r = this.unitDiscriminator - other.unitDiscriminator // compare ranges\n return if (isNegative()) -r else r\n }\n\n /**\n * Splits this duration into days, hours, minutes, seconds, and nanoseconds and executes the given [action] with these components.\n * The result of [action] is returned as the result of this function.\n *\n * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the whole number of seconds in this duration, and its absolute

```

```

value is less than 60;\n * - `minutes` represents the whole number of minutes in this duration, and its absolute
value is less than 60;\n * - `hours` represents the whole number
of hours in this duration, and its absolute value is less than 24;\n * - `days` represents the whole number of days
in this duration.\n * \n * Infinite durations are represented as either [Long.MAX_VALUE] days, or
[Long.MIN_VALUE] days (depending on the sign of infinity),\n * and zeroes in the lower components.\n * \n
public inline fun <T> toComponents(action: (days: Long, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -
> T): T {\n contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n return
action(inWholeDays, hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)\n } \n \n
/** \n * Splits this duration into hours, minutes, seconds, and nanoseconds and executes the given [action] with
these components.\n * The result of [action] is returned as the result of this function.\n * \n * - `nanoseconds`
represents the whole number of nanoseconds in this duration, and its absolute value is less
than 1_000_000_000;\n * - `seconds` represents the whole number of seconds in this duration, and its absolute
value is less than 60;\n * - `minutes` represents the whole number of minutes in this duration, and its absolute
value is less than 60;\n * - `hours` represents the whole number of hours in this duration.\n * \n * Infinite
durations are represented as either [Long.MAX_VALUE] hours, or [Long.MIN_VALUE] hours (depending on the
sign of infinity),\n * and zeroes in the lower components.\n * \n public inline fun <T> toComponents(action:
(hours: Long, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {\n contract { callsInPlace(action,
InvocationKind.EXACTLY_ONCE) }\n return action(inWholeHours, minutesComponent, secondsComponent,
nanosecondsComponent)\n } \n \n /** \n * Splits this duration into minutes, seconds, and nanoseconds and
executes the given [action] with these components.\n * The result of [action] is returned
as the result of this function.\n * \n * - `nanoseconds` represents the whole number of nanoseconds in this
duration, and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the whole number of
seconds in this duration, and its absolute value is less than 60;\n * - `minutes` represents the whole number of
minutes in this duration.\n * \n * Infinite durations are represented as either [Long.MAX_VALUE] minutes, or
[Long.MIN_VALUE] minutes (depending on the sign of infinity),\n * and zeroes in the lower components.\n * \n
public inline fun <T> toComponents(action: (minutes: Long, seconds: Int, nanoseconds: Int) -> T): T {\n
contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n return action(inWholeMinutes,
secondsComponent, nanosecondsComponent)\n } \n \n /** \n * Splits this duration into seconds, and
nanoseconds and executes the given [action] with these components.\n * The result of [action]
is returned as the result of this function.\n * \n * - `nanoseconds` represents the whole number of nanoseconds
in this duration, and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the whole number
of seconds in this duration.\n * \n * Infinite durations are represented as either [Long.MAX_VALUE] seconds,
or [Long.MIN_VALUE] seconds (depending on the sign of infinity),\n * and zero nanoseconds.\n * \n
public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T {\n contract {
callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n return action(inWholeSeconds,
nanosecondsComponent)\n } \n \n @PublishedApi\n internal val hoursComponent: Int\n get() = if
(isInfinite()) 0 else (inWholeHours % 24).toInt()\n \n @PublishedApi\n internal val minutesComponent: Int\n
get() = if (isInfinite()) 0 else (inWholeMinutes % 60).toInt()\n \n @PublishedApi\n internal val
secondsComponent:
Int\n get() = if (isInfinite()) 0 else (inWholeSeconds % 60).toInt()\n \n @PublishedApi\n internal val
nanosecondsComponent: Int\n get() = when {\n isInfinite() -> 0\n isInMillis() ->
millisToNanos(value % 1_000).toInt()\n else -> (value % 1_000_000_000).toInt()\n } \n \n //
conversion to units\n \n /** \n * Returns the value of this duration expressed as a [Double] number of the
specified [unit].\n * \n * The operation may involve rounding when the result cannot be represented exactly with
a [Double] number.\n * \n * An infinite duration value is converted either to [Double.POSITIVE_INFINITY] or
[Double.NEGATIVE_INFINITY] depending on its sign.\n * \n public fun toDouble(unit: DurationUnit):
Double {\n return when (rawValue) {\n INFINITE.rawValue -> Double.POSITIVE_INFINITY\n
NEG_INFİNITE.rawValue -> Double.NEGATIVE_INFINITY\n else -> {\n

```



```

// TODO: whether it's ok to convert to Double before scaling\n
convertDurationUnit(value.toDouble(), storageUnit, unit)\n }\n }\n }\n /**\n * Returns the value
of this duration expressed as a [Long] number of the specified [unit].\n *\n * If the result doesn't fit in the range
of [Long] type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n *\n * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n */\n public fun toLong(unit: DurationUnit): Long {\n return when (rawValue) {\n
INFINITE.rawValue -> Long.MAX_VALUE\n NEG_INFINITE.rawValue -> Long.MIN_VALUE\n
else -> convertDurationUnit(value, storageUnit, unit)\n }\n }\n /**\n * Returns the value of this
duration
expressed as an [Int] number of the specified [unit].\n *\n * If the result doesn't fit in the range of [Int] type, it
is coerced into that range:\n * - [Int.MIN_VALUE] is returned if it's less than `Int.MIN_VALUE`,\n * -
[Int.MAX_VALUE] is returned if it's greater than `Int.MAX_VALUE`.\n *\n * An infinite duration value is
converted either to [Int.MAX_VALUE] or [Int.MIN_VALUE] depending on its sign.\n */\n public fun
toInt(unit: DurationUnit): Int =\n toLong(unit).coerceIn(Int.MIN_VALUE.toLong(),
Int.MAX_VALUE.toLong()).toInt()\n /** The value of this duration expressed as a [Double] number of days.
*/\n @ExperimentalTime\n @Deprecated("Use inWholeDays property instead or convert toDouble(DAYS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.DAYS)"))\n
@DeprecatedSinceKotlin(warningSince = "1.5", errorSince = "1.8")\n public val inDays: Double get() =
toDouble(DurationUnit.DAYS)\n /** The value of
this duration expressed as a [Double] number of hours. */\n @ExperimentalTime\n @Deprecated("Use
inWholeHours property instead or convert toDouble(HOURS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.HOURS)"))\n @DeprecatedSinceKotlin(warningSince = "1.5",
errorSince = "1.8")\n public val inHours: Double get() = toDouble(DurationUnit.HOURS)\n /** The value of
this duration expressed as a [Double] number of minutes. */\n @ExperimentalTime\n @Deprecated("Use
inWholeMinutes property instead or convert toDouble(MINUTES) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MINUTES)"))\n @DeprecatedSinceKotlin(warningSince = "1.5",
errorSince = "1.8")\n public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)\n /** The
value of this duration expressed as a [Double] number of seconds. */\n @ExperimentalTime\n
@Deprecated("Use inWholeSeconds property instead or convert toDouble(SECONDS) if
a double value is required.", ReplaceWith("toDouble(DurationUnit.SECONDS)"))\n
@DeprecatedSinceKotlin(warningSince = "1.5", errorSince = "1.8")\n public val inSeconds: Double get() =
toDouble(DurationUnit.SECONDS)\n /** The value of this duration expressed as a [Double] number of
milliseconds. */\n @ExperimentalTime\n @Deprecated("Use inWholeMilliseconds property instead or convert
toDouble(MILLISECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MILLISECONDS)"))\n @DeprecatedSinceKotlin(warningSince = "1.5",
errorSince = "1.8")\n public val inMilliseconds: Double get() = toDouble(DurationUnit.MILLISECONDS)\n
/** The value of this duration expressed as a [Double] number of microseconds. */\n @ExperimentalTime\n
@Deprecated("Use inWholeMicroseconds property instead or convert toDouble(MICROSECONDS) if a double
value is required.", ReplaceWith("toDouble(DurationUnit.MICROSECONDS)"))\n
@DeprecatedSinceKotlin(warningSince
= "1.5", errorSince = "1.8")\n public val inMicroseconds: Double get() =
toDouble(DurationUnit.MICROSECONDS)\n /** The value of this duration expressed as a [Double] number of
nanoseconds. */\n @ExperimentalTime\n @Deprecated("Use inWholeNanoseconds property instead or convert
toDouble(NANOSECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.NANOSECONDS)"))\n @DeprecatedSinceKotlin(warningSince =
"1.5", errorSince = "1.8")\n public val inNanoseconds: Double get() =
toDouble(DurationUnit.NANOSECONDS)\n /**\n * The value of this duration expressed as a [Long]

```

```

number of days.\n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeDays: Long\n get() =
toLong(DurationUnit.DAYS)\n\n /**\n * The value of this duration expressed as a [Long] number of hours.\n
*\n * An infinite duration value
is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val
inWholeHours: Long\n get() = toLong(DurationUnit.HOURS)\n\n /**\n * The value of this duration
expressed as a [Long] number of minutes.\n * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeMinutes:
Long\n get() = toLong(DurationUnit.MINUTES)\n\n /**\n * The value of this duration expressed as a
[Long] number of seconds.\n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeSeconds: Long\n get() =
toLong(DurationUnit.SECONDS)\n\n /**\n * The value of this duration expressed as a [Long] number of
milliseconds.\n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n
*/\n public val inWholeMilliseconds: Long\n get() {\n return if (isInMillis() && isFinite()) value
else toLong(DurationUnit.MILLISECONDS)\n }\n\n /**\n * The value of this duration expressed as a
[Long] number of microseconds.\n * If the result doesn't fit in the range of [Long] type, it is coerced into
that range:\n * - [Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n * -
[Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n * An infinite duration value
is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val
inWholeMicroseconds: Long\n get() = toLong(DurationUnit.MICROSECONDS)\n\n /**\n * The value of
this duration expressed as a [Long] number of nanoseconds.\n * If the result doesn't fit in the range of
[Long] type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n
*/\n * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n */\n public val inWholeNanoseconds: Long\n get() {\n val value = value\n return
when {\n isInNanos() -> value\n value > Long.MAX_VALUE / NANOS_IN_MILLIS ->
Long.MAX_VALUE\n value < Long.MIN_VALUE / NANOS_IN_MILLIS -> Long.MIN_VALUE\n else -> millisToNanos(value)\n }\n }\n\n // shortcuts\n\n /**\n * Returns the value of this
duration expressed as a [Long] number of nanoseconds.\n * If the value doesn't fit in the range of [Long]
type, it is coerced into that range, see the conversion [Double.toLong] for details.\n * The range of
durations that can be expressed as a `Long` number of nanoseconds is approximately \u00b1292
years.\n */\n @ExperimentalTime\n @Deprecated("Use inWholeNanoseconds property instead.")\n
ReplaceWith("this.inWholeNanoseconds")\n @DeprecatedSinceKotlin(warningSince = "1.5", errorSince =
"1.8")\n public fun toLongNanoseconds(): Long = inWholeNanoseconds\n\n /**\n * Returns the value of
this duration expressed as a [Long] number of milliseconds.\n * The value is coerced to the range of [Long]
type, if it doesn't fit in that range, see the conversion [Double.toLong] for details.\n * The range of
durations that can be expressed as a `Long` number of milliseconds is approximately \u00b1292 million years.\n
*/\n @ExperimentalTime\n @Deprecated("Use inWholeMilliseconds property instead.")\n
ReplaceWith("this.inWholeMilliseconds")\n @DeprecatedSinceKotlin(warningSince = "1.5", errorSince =
"1.8")\n public fun toLongMilliseconds(): Long = inWholeMilliseconds\n\n /**\n * Returns a string
representation of this
duration value\n * expressed as a combination of numeric components, each in its own unit.\n * Each
component is a number followed by the unit abbreviated name: `d`, `h`, `m`, `s`:\n * `5h`, `1d 12h`, `1h 0m
30.340s`.\n * The last component, usually seconds, can be a number with a fractional part.\n * If the
duration is less than a second, it is represented as a single number\n * with one of sub-second units: `ms`
(milliseconds), `us` (microseconds), or `ns` (nanoseconds):\n * `140.884ms`, `500us`, `24ns`.\n * A
negative duration is prefixed with `-` sign and, if it consists of multiple components, surrounded with
parentheses:\n

```

```

* `~12m` and `~(1h 30m)`.\n * \n * Special cases:\n * - an infinite duration is formatted as `~Infinity` or
`~-Infinity` without a unit.\n * \n * It's recommended to use [toIsoString] that uses more strict ISO-8601
format instead of this `toString`\n * when you want to
convert a duration to a string in cases of serialization, interchange, etc.\n * \n * @sample
samples.time.Durations.toStringDefault\n * \n override fun toString(): String = when (rawValue) {\n OL ->
`0s`\n INFINITE.rawValue -> `Infinity`\n NEG_INFINITE.rawValue -> `~Infinity`\n else -> {\n
val isNegative = isNegative()\n buildString {\n if (isNegative) append('-')\n
absoluteValue.toComponents { days, hours, minutes, seconds, nanoseconds ->\n val hasDays = days !=
OL\n val hasHours = hours != 0\n val hasMinutes = minutes != 0\n val
hasSeconds = seconds != 0 || nanoseconds != 0\n var components = 0\n if (hasDays) {\n
append(days).append('d')\n components++\n }\n if (hasHours ||
(hasDays && (hasMinutes
|| hasSeconds))) {\n if (components++ > 0) append(' ')\n append(hours).append('h')\n
}\n if (hasMinutes || (hasSeconds && (hasHours || hasDays))) {\n if
(components++ > 0) append(' ')\n append(minutes).append('m')\n }\n if
(hasSeconds) {\n if (components++ > 0) append(' ')\n when {\n
seconds != 0 || hasDays || hasHours || hasMinutes ->\n appendFractional(seconds, nanoseconds, 9,
`s`, isoZeroes = false)\n nanoseconds >= 1_000_000 ->\n
appendFractional(nanoseconds / 1_000_000, nanoseconds % 1_000_000, 6, `ms`, isoZeroes = false)\n
nanoseconds >= 1_000 ->\n appendFractional(nanoseconds / 1_000, nanoseconds
% 1_000, 3, `us`, isoZeroes = false)\n else ->\n
append(nanoseconds).append(`ns`)\n }\n }\n if (isNegative && components
> 1) insert(1, (').append('))\n }\n }\n }\n }\n private fun
StringBuilder.appendFractional(whole: Int, fractional: Int, fractionalSize: Int, unit: String, isoZeroes: Boolean) {\n
append(whole)\n if (fractional != 0) {\n append('.')\n val fracString =
fractional.toString().padStart(fractionalSize, '0')\n val nonZeroDigits = fracString.indexOfLast { it != '0' } +
1\n when {\n !isoZeroes && nonZeroDigits < 3 -> appendRange(fracString, 0, nonZeroDigits)\n
else -> appendRange(fracString, 0, ((nonZeroDigits + 2) / 3) * 3)\n }\n }\n append(unit)\n
}\n\n /**\n * Returns a string representation
of this duration value expressed in the given [unit]\n * and formatted with the specified [decimals] number of
digits after decimal point.\n * \n * Special cases:\n * - an infinite duration is formatted as `~Infinity` or `~-
Infinity` without a unit.\n * \n * @param decimals the number of digits after decimal point to show. The value
must be non-negative.\n * \n * No more than 12 decimals will be shown, even if a larger number is requested.\n * \n
* @return the value of duration in the specified [unit] followed by that unit abbreviated name: `d`, `h`, `m`, `s`,
`ms`, `us`, or `ns`.\n * \n * @throws IllegalArgumentException if [decimals] is less than zero.\n * \n *
@sample samples.time.Durations.toStringDecimals\n * \n public fun toString(unit: DurationUnit, decimals: Int
= 0): String {\n require(decimals >= 0) { `decimals must be not negative, but was $decimals` }\n val
number = toDouble(unit)\n
if (number.isInfinite()) return number.toString()\n return formatToExactDecimals(number,
decimals.coerceAtMost(12)) + unit.shortName()\n }\n\n /**\n * Returns an ISO-8601 based string
representation of this duration.\n * \n * The returned value is presented in the format `PThHmMs.fS`, where `h`,
`m`, `s` are the integer components of this duration (see [toComponents])\n * and `f` is a fractional part of second.
Depending on the roundness of the value the fractional part can be formatted with either\n * 0, 3, 6, or 9 decimal
digits.\n * \n * The infinite duration is represented as `~PT999999999999H` which is larger than any
possible finite duration in Kotlin.\n * \n * Negative durations are indicated with the sign `~` in the beginning of
the returned string, for example, `~-PT5M30S`.\n * \n * @sample samples.time.Durations.toIsoString\n
* \n public fun toIsoString(): String = buildString {\n if (isNegative()) append('-')\n

```

```

 append("PT")\n this@Duration.absoluteValue.toComponents { hours, minutes, seconds, nanoseconds -
>\n @Suppress("NAME_SHADOWING")\n var hours = hours\n if (isInfinite()) {\n
// use large enough value instead of Long.MAX_VALUE\n hours = 9_999_999_999_999\n }\n
 val hasHours = hours != 0L\n val hasSeconds = seconds != 0 || nanoseconds != 0\n val hasMinutes =
minutes != 0 || (hasSeconds && hasHours)\n if (hasHours) {\n append(hours).append('H')\n
}\n if (hasMinutes) {\n append(minutes).append('M')\n }\n if (hasSeconds ||
(!hasHours && !hasMinutes)) {\n appendFractional(seconds, nanoseconds, 9, "S", isoZeroes = true)\n
}\n }\n }\n\n// constructing from number of units\n// extension functions\n\n/** Returns a [Duration]
equal to this

```

[Int] number of the specified [unit].

```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic fun Int.toDuration(unit:
DurationUnit): Duration {\n return if (unit <= DurationUnit.SECONDS) {\n
durationOfNanos(convertDurationUnitOverflow(this.toLong(), unit, DurationUnit.NANOSECONDS))\n } else\n
toLong().toDuration(unit)\n}\n\n/** Returns a [Duration] equal to this [Long] number of the specified [unit].

```

```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic fun Long.toDuration(unit:
DurationUnit): Duration {\n val maxNsInUnit = convertDurationUnitOverflow(MAX_NANOS,
DurationUnit.NANOSECONDS, unit)\n if (this in -maxNsInUnit..maxNsInUnit) {\n return
durationOfNanos(convertDurationUnitOverflow(this, unit, DurationUnit.NANOSECONDS))\n } else {\n val
millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS)\n return
durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n }\n}\n\n/**\n

```

\* Returns a [Duration] equal to this [Double] number of the specified [unit].\n \* Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n \* @throws

IllegalArgumentException if this `Double` value is `NaN`.\n

```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic fun Double.toDuration(unit:
DurationUnit): Duration {\n val valueInNs = convertDurationUnit(this, unit, DurationUnit.NANOSECONDS)\n
require(!valueInNs.isNaN()) { "Duration value cannot be NaN." }\n val nanos = valueInNs.roundToLong()\n
return if (nanos in -MAX_NANOS..MAX_NANOS) {\n durationOfNanos(nanos)\n } else {\n val millis =
convertDurationUnit(this, unit, DurationUnit.MILLISECONDS).roundToLong()\n
durationOfMillisNormalized(millis)\n }\n}\n\n// constructing from number of units\n// deprecated extension
properties\n\n/** Returns a [Duration] equal to this [Int] number of nanoseconds.

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Int.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"kotlin.time.Duration.Companion.nanoseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5", errorSince =
"1.8")\npublic val Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/** Returns a
[Duration] equal to this [Long] number of nanoseconds.

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.nanoseconds' extension property from
Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"kotlin.time.Duration.Companion.nanoseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5", errorSince =
"1.8")\npublic val Long.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/**\n * Returns a
[Duration] equal to this [Double] number of nanoseconds.\n * @throws IllegalArgumentException if this
[Double] value is `NaN`.\n *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use

```

```

'Double.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"kotlin.time.Duration.Companion.nanoseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5", errorSince =
"1.8")\npublic val Double.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/** Returns a
[Duration] equal to this [Int] number of microseconds.

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.microseconds' extension property from
Duration.Companion instead.", ReplaceWith("this.microseconds",
"kotlin.time.Duration.Companion.microseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5", errorSince

```

```

= \"1.8\")\npublic val Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a
[Duration] equal to this [Long] number of microseconds.
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.microseconds' extension property
from Duration.Companion instead.\", ReplaceWith(\"this.microseconds\",
\"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince
= \"1.5\", errorSince = \"1.8\")\npublic val Long.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a [Duration] equal to this [Double] number of
microseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.microseconds' extension property
from Duration.Companion instead.\", ReplaceWith(\"this.microseconds\",
\"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince
= \"1.8\")\npublic val Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a
[Duration] equal to this [Int] number of milliseconds.
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.milliseconds' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\",
\"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince
= \"1.5\", errorSince = \"1.8\")\npublic val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of
milliseconds. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.milliseconds' extension
property from Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\",
\"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a
[Duration] equal to this [Double] number of milliseconds.\n *\n * @throws IllegalArgumentException if this
[Double] value is `NaN`.\n *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Double.milliseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\",
\"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince
= \"1.5\", errorSince = \"1.8\")\npublic val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this [Int] number of seconds.
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.seconds' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.seconds\",
\"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to
this [Long] number of seconds. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Long.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.seconds\",
\"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration]
equal to this
[Double] number of seconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.seconds' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.seconds\",
\"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration]
equal to this [Int] number of minutes. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Int.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\",
\"kotlin.time.Duration.Companion.minutes\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to
this [Long] number of minutes. *\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use

```

```

'Long.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\",
\"kotlin.time.Duration.Companion.minutes\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration]
equal to this [Double] number of minutes.\n * \n * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n * \n * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.minutes' extension
property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\",
\"kotlin.time.Duration.Companion.minutes\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration]
equal to this [Int] number of hours. * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
'Int.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\",
\"kotlin.time.Duration.Companion.hours\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this
[Long] number of hours. * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.hours'
extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\",
\"kotlin.time.Duration.Companion.hours\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n/** \n * Returns a [Duration] equal to
this [Double] number of hours.\n * \n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
* \n * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.hours' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.hours\",
\"kotlin.time.Duration.Companion.hours\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic
val Double.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Int]
number of days. * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.days' extension
property from Duration.Companion instead.\", ReplaceWith(\"this.days\",
\"kotlin.time.Duration.Companion.days\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Int.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this
[Long] number of days. * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.days'
extension property from Duration.Companion instead.\", ReplaceWith(\"this.days\",
\"kotlin.time.Duration.Companion.days\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Long.days get() = toDuration(DurationUnit.DAYS)\n\n/** \n * Returns a [Duration] equal to this
[Double] number of days.\n * \n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
* \n * \n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.days' extension property from
Duration.Companion instead.\", ReplaceWith(\"this.days\",
\"kotlin.time.Duration.Companion.days\")\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\npublic val Double.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value
is the specified [duration] value multiplied by this number.
* \n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Int.times(duration: Duration): Duration = duration * this\n\n/** Returns a duration whose
value is the specified [duration] value multiplied by this number.\n * \n * The operation may involve rounding when
the result cannot be represented exactly with a [Double] number.\n * \n * @throws IllegalArgumentException if the
operation results in a `NaN` value.\n
* \n * \n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Double.times(duration: Duration): Duration = duration * this\n\n\nprivate fun
parseDuration(value: String, strictIso: Boolean): Duration {\n var length = value.length\n if (length == 0) throw
IllegalArgumentException(\"The string is empty\")\n var index = 0\n var result = Duration.ZERO\n val
infinityString = \"Infinity\"\n when (value[index]) {\n '+', '-' -> index++\n }\n val hasSign = index > 0\n val isNegative = hasSign && value.startsWith('-')\n when {\n length <= index ->\n throw
IllegalArgumentException(\"No components\")\n value[index] == 'P' -> {\n if (++index == length) throw

```

```

IllegalArgumentException()\n val nonDigitSymbols = \"+-.\n var isTimeComponent = false\n var prevUnit: DurationUnit? = null\n while (index < length) {\n if (value[index] == 'T') {\n if (isTimeComponent\n || ++index == length) throw IllegalArgumentException()\n isTimeComponent = true\n }\n continue\n }\n val component = value.substringWhile(index) { it in '0'..'9' || it in nonDigitSymbols }\n if (component.isEmpty()) throw IllegalArgumentException()\n index += component.length\n val unitChar = value.getOrElse(index) { throw IllegalArgumentException(\"Missing unit for value\n $component\") }\n index++\n val unit = durationUnitByIsoChar(unitChar, isTimeComponent)\n if (prevUnit != null && prevUnit <= unit) throw IllegalArgumentException(\"Unexpected order of duration\n components\")\n prevUnit = unit\n val dotIndex = component.indexOf('.')\n if (unit ==\n DurationUnit.SECONDS && dotIndex > 0) {\n val whole = component.substring(0, dotIndex)\n result += parseOverLongIsoComponent(whole).toDuration(unit)\n result += component.substring(dotIndex).toDouble().toDuration(unit)\n } else {\n result += parseOverLongIsoComponent(component).toDuration(unit)\n }\n }\n }\n strictIso\n -> throw IllegalArgumentException()\n value.regionMatches(index, infinityString, 0, length =\n maxOf(length - index, infinityString.length), ignoreCase = true) -> {\n result = Duration.INFINITE\n }\n else -> {\n // parse default string format\n var prevUnit: DurationUnit? = null\n var\n afterFirst = false\n var allowSpaces = !hasSign\n if (hasSign && value[index] == '(' && value.last()\n == ')') {\n allowSpaces = true\n if (++index == --length) throw IllegalArgumentException(\"No\n components\")\n }\n while (index < length) {\n if\n (afterFirst && allowSpaces) {\n index = value.skipWhile(index) { it == ' ' }\n }\n afterFirst = true\n val component = value.substringWhile(index) { it in '0'..'9' || it == ' ' }\n if\n (component.isEmpty()) throw IllegalArgumentException()\n index += component.length\n val\n unitName = value.substringWhile(index) { it in 'a'..'z' }\n index += unitName.length\n val unit =\n durationUnitByShortName(unitName)\n if (prevUnit != null && prevUnit <= unit) throw\n IllegalArgumentException(\"Unexpected order of duration components\")\n prevUnit = unit\n val\n dotIndex = component.indexOf('.')\n if (dotIndex > 0) {\n val whole = component.substring(0,\n dotIndex)\n result += whole.toLong().toDuration(unit)\n result +=\n component.substring(dotIndex).toDouble().toDuration(unit)\n }\n if (index < length) throw IllegalArgumentException(\"Fractional component must be last\")\n }\n } else {\n result += component.toLong().toDuration(unit)\n }\n }\n }\n }\n return if (isNegative) -result else result\n }\n\n private fun parseOverLongIsoComponent(value: String): Long {\n val length = value.length\n var startIndex = 0\n if (length > 0 && value[0] in \"+-") startIndex++\n if ((length\n - startIndex) > 16 && (startIndex..value.lastIndex).all { value[it] in '0'..'9' }) {\n // all chars are digits, but more\n than ceiling(log10(MAX_MILLIS / 1000)) of them\n return if (value[0] == '-') Long.MIN_VALUE else\n Long.MAX_VALUE\n }\n // TODO: replace with just toLong after min JDK becomes 8\n return if\n (value.startsWith(\"+\")) value.drop(1).toLong() else value.toLong()\n }\n\n private inline fun\n String.substringWhile(startIndex: Int, predicate:\n (Char) -> Boolean): String =\n substring(startIndex, skipWhile(startIndex, predicate))\n\n private inline fun\n String.skipWhile(startIndex: Int, predicate: (Char) -> Boolean): Int {\n var i = startIndex\n while (i < length &&\n predicate(this[i])) i++\n return i\n }\n\n // The ranges are chosen so that they are:\n // - symmetric relative to\n zero: this greatly simplifies operations with sign, e.g. unaryMinus and minus.\n // - non-overlapping, but adjacent:\n the first value that doesn't fit in nanos range, can be exactly represented in millis.\n\n internal const val\n NANOS_IN_MILLIS = 1_000_000\n // maximum number duration can store in nanosecond range\n\n internal const\n val MAX_NANOS = Long.MAX_VALUE / 2 / NANOS_IN_MILLIS * NANOS_IN_MILLIS - 1\n // ends in\n ..._999_999\n // maximum number duration can store in millisecond range, also encodes an infinite value\n\n internal\n const val MAX_MILLIS = Long.MAX_VALUE / 2\n // MAX_NANOS expressed in milliseconds\n\n private const val\n MAX_NANOS_IN_MILLIS

```

```

= MAX_NANOS / NANOS_IN_MILLIS\n\nprivate fun nanosToMillis(nanos: Long): Long = nanos /
NANOS_IN_MILLIS\n\nprivate fun millisToNanos(millis: Long): Long = millis * NANOS_IN_MILLIS\n\nprivate
fun durationOfNanos(normalNanos: Long) = Duration(normalNanos shl 1)\n\nprivate fun
durationOfMillis(normalMillis: Long) = Duration((normalMillis shl 1) + 1)\n\nprivate fun durationOf(normalValue:
Long, unitDiscriminator: Int) = Duration((normalValue shl 1) + unitDiscriminator)\n\nprivate fun
durationOfNanosNormalized(nanos: Long) =\n if (nanos in -MAX_NANOS..MAX_NANOS) {\n
durationOfNanos(nanos)\n } else {\n durationOfMillis(nanosToMillis(nanos))\n }\n\nprivate fun
durationOfMillisNormalized(millis: Long) =\n if (millis in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n durationOfNanos(millisToNanos(millis))\n }
else {\n durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n }\n\ninternal expect val
durationAssertionsEnabled: Boolean\n\ninternal expect
fun formatToExactDecimals(value: Double, decimals: Int): String\n\ninternal expect fun formatUpToDecimals(value:
Double, decimals: Int): String\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("UnsignedKt")\n\npackage
kotlin\n\n@PublishedApi\n\ninternal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor
Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)\n\n@PublishedApi\n\ninternal fun ulongCompare(v1: Long,
v2: Long): Int = (v1 xor Long.MIN_VALUE).compareTo(v2 xor Long.MIN_VALUE)\n\n@PublishedApi\n\ninternal
fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()\n\n@PublishedApi\n\ninternal fun
uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()\n\n// Division and remainder are
based on Guava's UnsignedLongs implementation\n\n// Copyright 2011 The Guava
Authors\n\n@PublishedApi\n\ninternal
fun ulongDivide(v1: ULong, v2: ULong): ULong {\n val dividend = v1.toLong()\n val divisor = v2.toLong()\n
if (divisor < 0) { // i.e., divisor >= 2^63:\n return if (v1 < v2) ULong(0) else ULong(1)\n }\n\n //
Optimization - use signed division if both dividend and divisor < 2^63\n if (dividend >= 0) {\n return
ULong(dividend / divisor)\n }\n\n // Otherwise, approximate the quotient, check, and correct if necessary.\n
val quotient = ((dividend ushr 1) / divisor) shl 1\n val rem = dividend - quotient * divisor\n return
ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)\n\n}\n\n@PublishedApi\n\ninternal fun
ulongRemainder(v1: ULong, v2: ULong): ULong {\n val dividend = v1.toLong()\n val divisor = v2.toLong()\n
if (divisor < 0) { // i.e., divisor >= 2^63:\n return if (v1 < v2) {\n v1 // dividend < divisor\n } else {\n
v1 - v2 // dividend >= divisor\n }\n }\n\n // Optimization - use signed modulus if both dividend and divisor < 2^63\n
if (dividend >= 0) {\n return
ULong(dividend % divisor)\n }\n\n // Otherwise, approximate the quotient, check, and correct if necessary.\n
val quotient = ((dividend ushr 1) / divisor) shl 1\n val rem = dividend - quotient * divisor\n return ULong(rem -
if (ULong(rem) >= ULong(divisor)) divisor else 0)\n\n}\n\n@PublishedApi\n\ninternal fun doubleToUInt(v: Double):
UInt = when {\n v.isNaN() -> 0u\n v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE\n v >=
UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE\n v <= Int.MAX_VALUE -> v.toInt().toUInt()\n else -
> (v - Int.MAX_VALUE).toInt().toUInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v <
UInt.MAX_VALUE\n}\n\n@PublishedApi\n\ninternal fun doubleToULong(v: Double): ULong = when {\n
v.isNaN() -> 0u\n v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE\n v >=
ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE\n v < Long.MAX_VALUE
-> v.toLong().toULong()\n\n // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not
representable in Double, so don't handle them.\n else -> (v - 9223372036854775808.0).toLong().toULong() +
9223372036854775808uL // Long.MAX_VALUE + 1 < v <
ULong.MAX_VALUE\n}\n\n\n\n@PublishedApi\n\ninternal fun uintToDouble(v: Int): Double = (v and
Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n\n@PublishedApi\n\ninternal fun
ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\n\n\n\ninternal fun
ulongToString(v: Long): String = ulongToString(v, 10)\n\n\n\ninternal fun ulongToString(v: Long, base: Int): String {\n

```



```

 if (v >= 0) return v.toString(base)\n\n var quotient = ((v ushr 1) / base) shl 1\n var rem = v - quotient * base\n if (rem >= base) {\n rem -= base\n quotient += 1\n }\n return quotient.toString(base) +
rem.toString(base)\n}\n\n", "/*\n * Copyright 2010-2016 JetBrains s.r.o.\n * \n * Licensed under the
Apache License, Version 2.0 (the "License");\n * you may not use this file except in compliance with the
License.\n * You may obtain a copy of the License at\n * \n * http://www.apache.org/licenses/LICENSE-2.0\n * \n *
Unless required by applicable law or agreed to in writing, software\n * distributed under the License is distributed
on an "AS IS" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.\n * See the License for the specific language governing permissions and\n * limitations under the
License.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding type parameter is not used for
unsafe operations such as casts or 'is' checks\n * That means it's completely safe to use generic types as argument for
such parameter.\n */\n\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal
annotation class PureReifiable\n\n/**\n * Specifies that the corresponding built-in method exists depending on
platform.\n */\n\n * Current implementation for JVM looks whether method with same JVM descriptor exists in the module JDK.\n *
For example MutableMap.remove(K, V) available only if corresponding\n * method
'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z' is defined in JDK (i.e. for major versions >= 8)\n
*\n\n@Target(AnnotationTarget.FUNCTION)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation
class PlatformDependent\n\n/**\n * When applied to a function or property, enables a compiler optimization that
evaluates that function or property\n * at compile-time and replaces calls to it with the computed result.\n */\n\n@Target(AnnotationTarget.CONSTRUCTOR, AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.7")\ninternal
annotation class IntrinsicConstEvaluation\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\n/**\n * Given an [iterator] function constructs an [Iterable] instance that returns values through
the [Iterator]\n * provided by that function.\n * @sample samples.collections.Iterables.Building.iterable\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T>
= object : Iterable<T> {\n override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * A wrapper over another
[Iterable] (or any other object that can produce an [Iterator]) that returns\n * an indexing iterator.\n */\n\ninternal class
IndexingIterable<out T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {\n override
fun iterator(): Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())\n}\n\n\n/**\n * Returns the size of
this iterable if it is known, or
`null` otherwise.\n */\n\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrNull(): Int? = if (this is
Collection<*>) this.size else null\n\n/**\n * Returns the size of this iterable if it is known, or the specified [default]
value otherwise.\n */\n\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrDefault(default: Int): Int = if
(this is Collection<*>) this.size else default\n\n\n/**\n * Returns a single list of all elements from all collections in
the given collection.\n * @sample samples.collections.Iterables.Operations.flattenIterable\n */\n\npublic fun <T>
Iterable<Iterable<T>>.flatten(): List<T> {\n val result = ArrayList<T>()\n for (element in this) {\n
result.addAll(element)\n }\n return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from
the first values of each pair from this collection,\n * *second* list is built from the second values of each pair from
this collection.\n * @sample samples.collections.Iterables.Operations.unzipIterable\n */\n\npublic fun <T, R> Iterable<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val expectedSize =
collectionSizeOrDefault(10)\n val listT = ArrayList<T>(expectedSize)\n val listR =
ArrayList<R>(expectedSize)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

```

license/LICENSE.txt file.\n

```
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\npackage
kotlin.sequences\n\nimport kotlin.random.Random\n\n/**\n * Given an [iterator] function constructs a [Sequence]
that returns values through the [Iterator]\n * provided by that function.\n * The values are evaluated lazily, and the
sequence is potentially infinite.\n */\n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>):
Sequence<T> = object : Sequence<T> {\n override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * Creates a
sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once.\n */\n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n *\npublic fun <T>
Iterator<T>.asSequence(): Sequence<T> = Sequence { this }.constrainOnce()\n\n/**\n * Creates a sequence that
returns the specified values.\n */\n * @sample samples.collections.Sequences.Building.sequenceOfValues\n *\npublic fun <T> sequenceOf(vararg elements: T): Sequence<T> = if (elements.isEmpty()) emptySequence() else
elements.asSequence()\n\n/**\n * Returns an empty sequence.\n */\n\npublic fun <T> emptySequence():
Sequence<T> = EmptySequence\n\nprivate object EmptySequence : Sequence<Nothing>,
DropTakeSequence<Nothing> {\n override fun iterator():
Iterator<Nothing> = EmptyIterator\n override fun drop(n: Int) = EmptySequence\n override fun take(n: Int) =
EmptySequence\n}\n\n/**\n * Returns this sequence if it's not `null` and the empty sequence otherwise.\n */\n * @sample samples.collections.Sequences.Usage.sequenceOrEmpty\n *\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>?.orEmpty():
Sequence<T> = this ?: emptySequence()\n\n/**\n * Returns a sequence that iterates through the elements either of
this sequence\n * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.\n */\n * @sample samples.collections.Sequences.Usage.sequenceIfEmpty\n *\n\n@SinceKotlin("1.3")\npublic fun
<T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {\n val iterator =
this@ifEmpty.iterator()\n if (iterator.hasNext()) {\n yieldAll(iterator)\n } else {\n yieldAll(defaultValue())\n }\n}\n\n/**\n * Returns a sequence
of all elements from all sequences in this sequence.\n */\n * The operation is _intermediate_ and _stateless_.\n */\n * @sample samples.collections.Sequences.Transformations.flattenSequenceOfSequences\n *\npublic fun <T>
Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }\n\n/**\n * Returns a sequence of all
elements from all iterables in this sequence.\n */\n * The operation is _intermediate_ and _stateless_.\n */\n * @sample samples.collections.Sequences.Transformations.flattenSequenceOfLists\n *\n\n@kotlin.jvm.JvmName("flattenSequenceOfIterable")\npublic fun <T> Sequence<Iterable<T>>.flatten():
Sequence<T> = flatten { it.iterator() }\n\nprivate fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>):
Sequence<R> {\n if (this is TransformingSequence<*, *>) {\n return (this as TransformingSequence<*,
T>).flatten(iterator)\n }\n return FlatteningSequence(this, { it }, iterator)\n}\n\n/**\n * Returns a pair of lists,
where\n * *first* list
is built from the first values of each pair from this sequence,\n * *second* list is built from the second values of
each pair from this sequence.\n */\n * The operation is _terminal_.\n */\n * @sample
samples.collections.Sequences.Transformations.unzip\n *\npublic fun <T, R> Sequence<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n val listT = ArrayList<T>()\n val listR = ArrayList<R>()\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n/**\n * Returns a sequence that
yields elements of this sequence randomly shuffled.\n */\n * Note that every iteration of the sequence returns
elements in a different order.\n */\n * The operation is _intermediate_ and _stateful_.\n *\n\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.shuffled(): Sequence<T> = shuffled(Random)\n\n/**\n * Returns a sequence that yields elements of this sequence randomly shuffled\n * using the specified [random]
instance as the source of randomness.\n *\n * Note that every iteration of the sequence returns elements in a different order.\n */\n * The operation is
intermediate and _stateful_.\n *\n\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.shuffled(random:
Random): Sequence<T> = sequence<T> {\n val buffer = toMutableList()\n while (buffer.isNotEmpty()) {\n
```

```

val j = random.nextInt(buffer.size)\n val last = buffer.removeLast()\n val value = if (j < buffer.size)
buffer.set(j, last) else last\n yield(value)\n } \n } \n\n/**\n * A sequence that returns the values from the
underlying [sequence] that either match or do not match\n * the specified [predicate].\n *\n * @param sendWhen If
`true`, values for which the predicate returns `true` are returned. Otherwise,\n * values for which the predicate
returns `false` are returned\n */\n\ninternal class FilteringSequence<T>(\n private val sequence: Sequence<T>,\n private val sendWhen: Boolean = true,\n private val predicate: (T) -> Boolean)\n
: Sequence<T> {\n\n override fun iterator(): Iterator<T> = object : Iterator<T> {\n val iterator =
sequence.iterator()\n var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n var nextItem: T?
= null\n\n private fun calcNext() {\n while (iterator.hasNext()) {\n val item = iterator.next()\n
 if (predicate(item) == sendWhen) {\n nextItem = item\n nextState = 1\n
 }\n }\n nextState = 0\n }\n\n override fun next(): T {\n if (nextState
== -1)\n calcNext()\n if (nextState == 0)\n throw NoSuchElementException()\n val
result = nextItem\n nextItem = null\n nextState = -1\n @Suppress("UNCHECKED_CAST")\n
 return result as T\n }\n\n override fun hasNext(): Boolean {\n if (nextState == -1)\n calcNext()\n return nextState == 1\n }\n }\n} \n\n/**\n * A
sequence which returns the results of applying the given [transformer] function to the values\n * in the underlying
[sequence].\n */\n\ninternal class TransformingSequence<T, R>(\n constructor(private val sequence: Sequence<T>,\n private val transformer: (T) -> R) : Sequence<R> {\n override fun iterator(): Iterator<R> = object : Iterator<R> {\n
 val iterator = sequence.iterator()\n override fun next(): R {\n return transformer(iterator.next())\n
 }\n\n override fun hasNext(): Boolean {\n return iterator.hasNext()\n }\n }\n\n internal fun <E>
flatten(iterator: (R) -> Iterator<E>): Sequence<E> {\n return FlatteningSequence<T, R, E>(sequence,\n transformer, iterator)\n }\n }\n} \n\n/**\n * A sequence which returns the results of applying the given [transformer]
function to the values\n * in the underlying
[sequence], where the transformer function takes the index of the value in the underlying\n * sequence along with
the value itself.\n */\n\ninternal class TransformingIndexedSequence<T, R>(\n constructor(private val sequence:
Sequence<T>,\n private val transformer: (Int, T) -> R) : Sequence<R> {\n override fun iterator(): Iterator<R> =
object : Iterator<R> {\n val iterator = sequence.iterator()\n var index = 0\n override fun next(): R {\n return transformer(checkIndexOverflow(index++), iterator.next())\n
 }\n\n override fun hasNext():
Boolean {\n return iterator.hasNext()\n }\n }\n }\n} \n\n/**\n * A sequence which combines values from
the underlying [sequence] with their indices and returns them as\n * [IndexedValue] objects.\n */\n\ninternal class
IndexingSequence<T>(\n constructor(private val sequence: Sequence<T>) : Sequence<IndexedValue<T>> {\n override fun iterator(): Iterator<IndexedValue<T>> =
object : Iterator<IndexedValue<T>>
{\n val iterator = sequence.iterator()\n var index = 0\n override fun next(): IndexedValue<T> {\n return IndexedValue(checkIndexOverflow(index++), iterator.next())\n
 }\n\n override fun hasNext():
Boolean {\n return iterator.hasNext()\n }\n }\n }\n} \n\n/**\n * A sequence which takes the values from
two parallel underlying sequences, passes them to the given\n * [transform] function and returns the values returned
by that function. The sequence stops returning\n * values as soon as one of the underlying sequences stops returning
values.\n */\n\ninternal class MergingSequence<T1, T2, V>(\n constructor(\n private val sequence1:
Sequence<T1>,\n private val sequence2: Sequence<T2>,\n private val transform: (T1, T2) -> V)\n) : Sequence<V> {\n override fun iterator(): Iterator<V> = object : Iterator<V> {\n val iterator1 =
sequence1.iterator()\n val iterator2 = sequence2.iterator()\n override
fun next(): V {\n return transform(iterator1.next(), iterator2.next())\n }\n\n override fun hasNext():
Boolean {\n return iterator1.hasNext() && iterator2.hasNext()\n }\n }\n }\n} \n\ninternal class
FlatteningSequence<T, R, E>(\n constructor(\n private val sequence: Sequence<T>,\n private val transformer: (T)
-> R,\n private val iterator: (R) -> Iterator<E>)\n) : Sequence<E> {\n override fun iterator(): Iterator<E> = object
: Iterator<E> {\n val iterator = sequence.iterator()\n var itemIterator: Iterator<E>? = null\n override
fun next(): E {\n if (!ensureItemIterator())\n throw NoSuchElementException()\n return
itemIterator!!.next()\n }\n\n override fun hasNext(): Boolean {\n return ensureItemIterator()\n
 }\n }\n }\n}

```

```

}\n\n private fun ensureItemIterator(): Boolean {\n if (itemIterator?.hasNext() == false)\nitemIterator = null\n\n while (itemIterator == null) {\n if (!iterator.hasNext()) {\n return false\n } else\n\n {\n val element = iterator.next()\n val nextItemIterator = iterator(transformer(element))\n if (nextItemIterator.hasNext()) {\n itemIterator = nextItemIterator\n return true\n }\n }\n }\n return true\n }\n}\n\ninternal fun <T, C, R>\nflatMapIndexed(source: Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> =\nsequence {\n var index = 0\n for (element in source) {\n val result =\ntransform(checkIndexOverflow(index++), element)\n yieldAll(iterator(result))\n }\n}\n\n/**\n * A\nsequence that supports drop(n) and take(n) operations\n */\ninternal interface DropTakeSequence<T> :\nSequence<T> {\n\n fun drop(n: Int): Sequence<T>\n fun take(n: Int): Sequence<T>\n}\n\n/**\n * A sequence that skips [startIndex]\nvalues from the underlying [sequence]\n * and stops returning values right before [endIndex], i.e. stops at `endIndex\n- 1`\n */\ninternal class SubSequence<T>(\n private val sequence: Sequence<T>,\n private val startIndex: Int,\nprivate val endIndex: Int\n) : Sequence<T>, DropTakeSequence<T> {\n\n init {\n require(startIndex >= 0) {\n \"startIndex should be non-negative, but is $startIndex\"\n }\n require(endIndex >= 0) {\n \"endIndex should be\nnon-negative, but is $endIndex\"\n }\n require(endIndex >= startIndex) {\n \"endIndex should be not less than\nstartIndex, but was $endIndex < $startIndex\"\n }\n }\n\n private val count: Int get() = endIndex - startIndex\n\n override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex\n+ n, endIndex)\n\n override fun take(n: Int): Sequence<T> = if (n >=\ncount) this else SubSequence(sequence, startIndex, startIndex + n)\n\n override fun iterator() = object :\nIterator<T> {\n\n val iterator = sequence.iterator()\n var position = 0\n\n // Shouldn't be called from\nconstructor to avoid premature iteration\n private fun drop() {\n while (position < startIndex &&\niterator.hasNext()) {\n iterator.next()\n position++\n }\n }\n\n override fun\nhasNext(): Boolean {\n drop()\n return (position < endIndex) && iterator.hasNext()\n }\n\n override fun next(): T {\n drop()\n if (position >= endIndex)\n throw\nNoSuchElementException()\n position++\n return iterator.next()\n }\n }\n}\n\n/**\n * A\nsequence that returns at most [count] values from the underlying [sequence], and stops returning values\n * as soon\nas that count is reached.\n */\ninternal class\nTakeSequence<T>(\n private val sequence: Sequence<T>,\n private val count: Int\n) : Sequence<T>,\nDropTakeSequence<T> {\n\n init {\n require(count >= 0) {\n \"count must be non-negative, but was $count.\"\n }\n }\n\n override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else\nSubSequence(sequence, n, count)\n\n override fun take(n: Int): Sequence<T> = if (n >= count) this else\nTakeSequence(sequence, n)\n\n override fun iterator(): Iterator<T> = object : Iterator<T> {\n var left =\ncount\n val iterator = sequence.iterator()\n\n override fun next(): T {\n if (left == 0)\n throw\nNoSuchElementException()\n left--\n return iterator.next()\n }\n\n override fun hasNext():\nBoolean {\n return left > 0 && iterator.hasNext()\n }\n }\n}\n\n/**\n * A sequence that returns values\nfrom the underlying [sequence] while the [predicate] function returns `true`,\nand stops returning values once the function returns `false` for the next element.\n */\ninternal class\nTakeWhileSequence<T>(\n private val sequence: Sequence<T>,\n private val predicate: (T) ->\nBoolean\n) : Sequence<T> {\n\n override fun iterator(): Iterator<T> = object : Iterator<T> {\n val iterator =\nsequence.iterator()\n var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n var nextItem: T?\n= null\n\n private fun calcNext() {\n if (iterator.hasNext()) {\n val item = iterator.next()\n if (predicate(item)) {\n nextState = 1\n nextItem = item\n return\n }\n nextState = 0\n }\n }\n\n override fun next(): T {\n if (nextState == -1)\n calcNext()\n // will change nextState\n if (nextState == 0)\n throw NoSuchElementException()\n @SuppressWarnings(\"UNCHECKED_CAST\")\n val result = nextItem as T\n // Clean next to avoid\nkeeping reference on yielded instance\n nextItem = null\n nextState = -1\n return result\n }\n }\n}

```

```

}\n\n override fun hasNext(): Boolean {\n if (nextState == -1)\n calcNext() // will change\n nextState\n return nextState == 1\n }\n}\n}\n\n/**\n * A sequence that skips the specified number of\n values from the underlying [sequence] and returns\n * all values after that.\n */\n\ninternal class DropSequence<T>(\n private val sequence: Sequence<T>,\n private val count: Int\n) : Sequence<T>, DropTakeSequence<T> {\n init\n {\n require(count >= 0) { \"count must be non-negative, but was $count.\" }\n }\n\n override fun drop(n:\n Int): Sequence<T> = (count + n).let { n1 -> if (n1 < 0) DropSequence(this, n) else DropSequence(sequence, n1) }\n\n override fun take(n: Int): Sequence<T>\n = (count + n).let { n1 -> if (n1 < 0) TakeSequence(this, n) else SubSequence(sequence, count, n1) }\n\n override\n fun iterator(): Iterator<T> = object : Iterator<T> {\n val iterator = sequence.iterator()\n var left = count\n\n // Shouldn't be called from constructor to avoid premature iteration\n private fun drop() {\n while (left\n > 0 && iterator.hasNext()) {\n iterator.next()\n left--\n }\n }\n\n override fun\n next(): T {\n drop()\n return iterator.next()\n }\n\n override fun hasNext(): Boolean {\n drop()\n return iterator.hasNext()\n }\n }\n}\n\n/**\n * A sequence that skips the values from the\n underlying [sequence] while the given [predicate] returns `true` and returns\n * all values after that.\n */\n\ninternal class DropWhileSequence<T>(\n private val sequence: Sequence<T>,\n private val predicate:\n (T) -> Boolean\n) : Sequence<T> {\n override fun iterator(): Iterator<T> = object : Iterator<T> {\n val\n iterator = sequence.iterator()\n var dropState: Int = -1 // -1 for not dropping, 1 for nextItem, 0 for normal\n iteration\n var nextItem: T? = null\n\n private fun drop() {\n while (iterator.hasNext()) {\n val item = iterator.next()\n if (!predicate(item)) {\n nextItem = item\n dropState =\n 1\n return\n }\n dropState = 0\n }\n\n override fun next(): T {\n if (dropState == -1)\n drop()\n if (dropState == 1)\n @Suppress(\"UNCHECKED_CAST\")\n val result = nextItem as T\n nextItem = null\n dropState = 0\n return result\n }\n return iterator.next()\n }\n\n override\n fun hasNext(): Boolean {\n if (dropState == -1)\n drop()\n return dropState == 1 ||\n iterator.hasNext()\n }\n }\n}\n\ninternal class DistinctSequence<T, K>(\n private val source: Sequence<T>,\n private val keySelector: (T) -> K\n) : Sequence<T> {\n override fun iterator(): Iterator<T> =\n DistinctIterator(source.iterator(), keySelector)\n}\n\nprivate class DistinctIterator<T, K>(\n private val source:\n Iterator<T>, private val keySelector: (T) -> K\n) : AbstractIterator<T>() {\n private val observed =\n HashSet<K>()\n override fun computeNext() {\n while (source.hasNext()) {\n val next =\n source.next()\n val key = keySelector(next)\n if (observed.add(key)) {\n setNext(next)\n return\n }\n done()\n }\n }\n}\n\nprivate class GeneratorSequence<T : Any>(\n private val\n getInitialValue: () -> T?, private val getNextValue: (T) -> T?> : Sequence<T>\n {\n override fun iterator(): Iterator<T> = object : Iterator<T> {\n var nextItem: T? = null\n var nextState:\n Int = -2 // -2 for initial unknown, -1 for next unknown, 0 for done, 1 for continue\n\n private fun calcNext() {\n nextItem = if (nextState == -2) getInitialValue() else getNextValue(nextItem!)\n nextState = if\n (nextItem == null) 0 else 1\n }\n\n override fun next(): T {\n if (nextState < 0)\n calcNext()\n\n if (nextState == 0)\n throw NoSuchElementException()\n val result =\n nextItem as T\n // Do not clean nextItem (to avoid keeping reference on yielded instance) -- need to keep\n state for getNextValue\n nextState = -1\n return result\n }\n\n override fun hasNext():\n Boolean {\n if (nextState < 0)\n calcNext()\n return nextState == 1\n }\n }\n}\n\n/**\n * Returns a\n wrapper sequence that provides values of this sequence, but ensures it can be iterated only one time.\n */\n\n/**\n * The\n operation is _intermediate_ and _stateless_.\n */\n\n/**\n * [IllegalStateException] is thrown on iterating the returned\n sequence for the second time and the following times.\n */\n\npublic fun <T> Sequence<T>.constrainOnce():\n Sequence<T> {\n // as? does not work in js\n //return this as? ConstrainedOnceSequence<T> ?:\n ConstrainedOnceSequence(this)\n}\n\nreturn if (this is ConstrainedOnceSequence<T>) this else\n ConstrainedOnceSequence(this)\n}\n\n/**\n * Returns a sequence which invokes the function to calculate the next\n value on each iteration until the function returns `null`.\n */\n\n/**\n * The returned sequence is constrained to be iterated

```

```

only once.\n *\n * @see constrainOnce\n *\n * @see kotlin.sequences.sequence\n *\n * @sample
samples.collections.Sequences.Building.generateSequence\n *\npublic fun <T : Any>
generateSequence(nextFunction: () -> T?): Sequence<T> {\n return GeneratorSequence(nextFunction,
 { nextFunction() }).constrainOnce()\n}\n\n/**\n * Returns a sequence defined by the starting value [seed] and the
function [nextFunction],\n * which is invoked to calculate the next value based on the previous one on each
iteration.\n *\n * The sequence produces values until it encounters first `null` value.\n * If [seed] is `null`, an empty
sequence is produced.\n *\n * The sequence can be iterated multiple times, each time starting with [seed].\n *\n *
@see kotlin.sequences.sequence\n *\n * @sample
samples.collections.Sequences.Building.generateSequenceWithSeed\n
*\n@kotlin.internal.LowPriorityInOverloadResolution\npublic fun <T : Any> generateSequence(seed: T?,
nextFunction: (T) -> T?): Sequence<T> =\n if (seed == null)\n EmptySequence\n else\n GeneratorSequence({ seed }, nextFunction)\n\n/**\n * Returns a sequence defined by the function [seedFunction],
which is invoked to produce the starting value,\n * and the [nextFunction], which
is invoked to calculate the next value based on the previous one on each iteration.\n *\n * The sequence produces
values until it encounters first `null` value.\n * If [seedFunction] returns `null`, an empty sequence is produced.\n *\n
* The sequence can be iterated multiple times.\n *\n * @see kotlin.sequences.sequence\n *\n * @sample
samples.collections.Sequences.Building.generateSequenceWithLazySeed\n *\npublic fun <T : Any>
generateSequence(seedFunction: () -> T?, nextFunction: (T) -> T?): Sequence<T> =\n GeneratorSequence(seedFunction, nextFunction)\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("PreconditionsKt")\n\npackage
kotlin\n\nimport kotlin.contracts.contract\n\n/**\n * Throws an [IllegalArgumentException] if the [value] is false.\n
*\n * @sample samples.misc.Preconditions.failRequireWithLazyMessage\n
*\n@kotlin.internal.InlineOnly\npublic inline fun require(value: Boolean): Unit {\n contract {\n returns()
implies value\n }\n require(value) { "Failed requirement." }\n}\n\n/**\n * Throws an
[IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is false.\n *\n * @sample
samples.misc.Preconditions.failRequireWithLazyMessage\n *\n@kotlin.internal.InlineOnly\npublic inline fun
require(value: Boolean, lazyMessage: () -> Any): Unit {\n contract {\n returns() implies value\n }\n if
(!value) {\n val message = lazyMessage()\n throw IllegalArgumentException(message.toString())\n }\n}\n\n/**\n * Throws an [IllegalArgumentException] if the [value] is null. Otherwise returns the not null value.\n
*\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> requireNotNull(value: T?): T {\n contract {\n
returns() implies (value != null)\n }\n return requireNotNull(value) { "Required value was null." }\n}\n\n/**\n * Throws an
[IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is null. Otherwise\n * returns the
not null value.\n *\n * @sample samples.misc.Preconditions.failRequireNotNullWithLazyMessage\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> requireNotNull(value: T?, lazyMessage: () -> Any): T
{\n contract {\n returns() implies (value != null)\n }\n if (value == null) {\n val message =
lazyMessage()\n throw IllegalArgumentException(message.toString())\n } else {\n return value\n }\n}\n\n/**\n * Throws an [IllegalStateException] if the [value] is false.\n *\n * @sample
samples.misc.Preconditions.failCheckWithLazyMessage\n *\n@kotlin.internal.InlineOnly\npublic inline fun
check(value: Boolean): Unit {\n contract {\n returns() implies value\n }\n check(value) { "Check failed."
}\n}\n\n/**\n * Throws
an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is false.\n *\n * @sample
samples.misc.Preconditions.failCheckWithLazyMessage\n *\n@kotlin.internal.InlineOnly\npublic inline fun
check(value: Boolean, lazyMessage: () -> Any): Unit {\n contract {\n returns() implies value\n }\n if
(!value) {\n val message = lazyMessage()\n throw IllegalStateException(message.toString())\n }\n}\n\n/**\n * Throws an [IllegalStateException] if the [value] is null. Otherwise\n * returns the not null value.\n

```

```

*\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T : Any> checkNotNull(value: T?): T {\n contract {\n returns() implies (value != null)\n }\n
return checkNotNull(value) { \"Required value was null.\" }\n}\n\n/**\n * Throws an [IllegalStateException] with
the result of calling [lazyMessage] if the [value] is null. Otherwise\n * returns the not null value.\n
*\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T : Any> checkNotNull(value: T?, lazyMessage: () -> Any): T {\n contract {\n returns() implies
(value != null)\n }\n\n if (value == null) {\n val message = lazyMessage()\n throw
IllegalStateException(message.toString())\n } else {\n return value\n }\n}\n\n\n/**\n * Throws an
[IllegalStateException] with the given [message].\n *\n * @sample samples.misc.Preconditions.failWithError\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun error(message: Any): Nothing = throw
IllegalStateException(message.toString())\n\"\", \"/>\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n//
See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.js.*\nimport
primitiveArrayConcat\nimport withType\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n *
Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds
of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun <T>
Array<out T>.elementAt(index: Int): T {\n return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\npublic actual fun ByteArray.elementAt(index: Int): Byte
{\n return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n}\n\n\n/**\n * Returns an element
at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n
*\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun
ShortArray.elementAt(index: Int): Short {\n return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n}\n\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\npublic actual fun IntArray.elementAt(index: Int): Int {\n
return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n}\n\n\n/**\n
* Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds
of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun
LongArray.elementAt(index: Int): Long {\n
return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n}\n\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic
actual fun FloatArray.elementAt(index: Int): Float {\n return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n}\n\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\npublic actual fun DoubleArray.elementAt(index: Int):
Double {\n return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n}\n\n\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this
array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun
BooleanArray.elementAt(index: Int): Boolean {\n return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n}\n\n\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample

```





```

*\n@SinceKotlin("1.4")\n@library("arrayDeepHashCode")\npublic actual fun <T> Array<out
T>?.contentDeepHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a string representation of the
contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains
itself on any nesting level that reference\n * is rendered as `"[...]"` to prevent recursion.\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentDeepToString\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic
actual fun <T> Array<out T>.contentDeepToString(): String {\n return this.contentDeepToString()\n}\n\n/**\n *
Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists
too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `"[...]"` to prevent
recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
*\n@SinceKotlin("1.4")\n@library("arrayDeepToString")\npublic actual fun <T> Array<out
T>?.contentDeepToString(): String {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays
are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n *
\n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers
it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun <T>
Array<out T>.contentEquals(other: Array<out T>): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
ByteArray.contentEquals(other: ByteArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n *
\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for
equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself
and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
ShortArray.contentEquals(other: ShortArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince
= "1.4")\npublic actual infix fun IntArray.contentEquals(other: IntArray): Boolean {\n return
this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one
another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are
compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is
equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
LongArray.contentEquals(other: LongArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with
the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0`
is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
FloatArray.contentEquals(other: FloatArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns

```

`true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun
DoubleArray.contentEquals(other: DoubleArray): Boolean {
 return this.contentEquals(other)
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun
BooleanArray.contentEquals(other: BooleanArray): Boolean {
 return this.contentEquals(other)
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun
CharArray.contentEquals(other: CharArray): Boolean {
 return this.contentEquals(other)
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun <T> Array<out
T>?.contentEquals(other: Array<out T>?): Boolean {
 definedExternally
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun ByteArray?.contentEquals(other:
ByteArray?): Boolean {
 definedExternally
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun ShortArray?.contentEquals(other:
ShortArray?): Boolean {
 definedExternally
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun IntArray?.contentEquals(other:
IntArray?): Boolean {
 definedExternally
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun LongArray?.contentEquals(other:
LongArray?): Boolean {
 definedExternally
}

```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun

```

FloatArray?.contentEquals(other: FloatArray?): Boolean {  
 definedExternally  
 Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.  
 The elements are compared for equality with the [equals][Any.equals] function.  
 For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```

@SinceKotlin("1.4")@library("arrayEquals")public actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {
 definedExternally
 Returns `true` if the two specified arrays are structurally equal to one another, i.e. contain the same number of the same elements in the same order.
 The elements are compared for equality with the [equals][Any.equals] function.
 For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
}

@SinceKotlin("1.4")@library("arrayEquals")public actual infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean {
 definedExternally
 Returns `true` if the two specified arrays are structurally equal to one another, i.e. contain the same number of the same elements in the same order.
 The elements are compared for equality with the [equals][Any.equals] function.
 For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
}

@SinceKotlin("1.4")@library("arrayEquals")public actual infix fun CharArray?.contentEquals(other: CharArray?): Boolean {
 definedExternally
 Returns a hash code based on the contents of this array as if it is [List].
}

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun <T> Array<out T>.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun ByteArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun ShortArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun IntArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun LongArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun FloatArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun DoubleArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun BooleanArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1")@DeprecatedSinceKotlin(hiddenSince = "1.4")public actual fun CharArray.contentHashCode(): Int {
 return this.contentHashCode()
}

Returns a hash code based on the contents of this array as if it is [List].

@SinceKotlin("1.4")@library("arrayHashCode")public actual fun <T> Array<out T>?.contentHashCode(): Int {
 definedExternally
 Returns a hash code based on the

```

contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 ByteArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the contents  
 of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 ShortArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the  
 contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 IntArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the contents  
 of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic  
 actual fun LongArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on  
 the contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual  
 fun FloatArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the  
 contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 DoubleArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the  
 contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 BooleanArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a hash code based on the  
 contents of this array as if it is [List].\n \*^@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun  
 CharArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/\*\*\n \* Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic  
 actual fun <T> Array<out T>.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a  
 string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic  
 actual fun ByteArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a string  
 representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation  
 warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun  
 ShortArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a string  
 representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic  
 actual fun IntArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a string  
 representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince  
 = "1.4")\npublic actual fun LongArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \*  
 Returns a string representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic  
 actual fun FloatArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a string  
 representation of the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4  
 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic  
 actual fun DoubleArray.contentToString(): String {\n return this.contentToString()\n}\n\n/\*\*\n \* Returns a string  
 representation of  
 the contents of the specified array as if it is [List].\n \* \n \* @sample  
 samples.collections.Arrays.ContentOperations.contentToString\n \*^@Deprecated("Use Kotlin compiler 1.4 to

```

avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun BooleanArray.contentToString(): String {\n return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.\")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun CharArray.contentToString(): String {\n return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic
actual fun <T> Array<out T>?.contentToString(): String {\n definedExternally\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ByteArray?.contentToString(): String
{\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ShortArray?.contentToString(): String
{\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic
actual fun IntArray?.contentToString(): String {\n definedExternally\n}\n\n/**\n * Returns a string representation
of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun LongArray?.contentToString(): String
{\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun FloatArray?.contentToString(): String
{\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun DoubleArray?.contentToString():
String {\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun BooleanArray?.contentToString():
String {\n definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*/\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun CharArray?.contentToString(): String
{\n definedExternally\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that
array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it
overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *
@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the
beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n
*\n * @return the [destination] array.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset:
Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {\n arrayCopy(this,

```

```

destination, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its
subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the
[destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param
destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by
default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param
endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out
of the [destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): ByteArray {\n arrayCopy(this.unsafeCast<Array<Byte>>(),
destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n
 * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n *
@param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to
copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n *
@param endIndex the
end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the
[destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int =
0, startIndex: Int = 0, endIndex: Int = size): ShortArray {\n arrayCopy(this.unsafeCast<Array<Short>>(),
destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n
 * Copies this array or its subrange into the [destination]
array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the
subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *
@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the
beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n
*\n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic
actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex:
Int = size): IntArray {\n arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(),
destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into
the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even
specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy
to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param
startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive)
of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or

```

[IllegalArgumentException] when [startIndex]

or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws

IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray {\n arrayCopy(this.unsafeCast<Array<Long>>(), destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray {\n arrayCopy(this.unsafeCast<Array<Float>>(), destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {\n arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {\n arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy
```

```
to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
```

```
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {\n arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy
```

```

Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {
arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(),
destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**
 * Copies this array or its subrange into
the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even
specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy
to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param
startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive)
of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n * or when that index is out of the [destination]
array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): CharArray {
arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress("ACTUAL_WITHOUT_EXPECT",
"NOTHING_TO_INLINE")\npublic actual inline fun <T> Array<out T>.copyOf(): Array<T> {
return
this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun
ByteArray.copyOfOf(): ByteArray {
return this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a
copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun ShortArray.copyOfOf(): ShortArray {
return this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n *
@sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun IntArray.copyOfOf(): IntArray {
return
this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\npublic actual fun LongArray.copyOfOf(): LongArray {
return withType("LongArray", this.asDynamic().slice())\n}\n\n/**
 * Returns new array which is a copy of the
original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun FloatArray.copyOfOf(): FloatArray {
return
this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n *
@sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun DoubleArray.copyOfOf(): DoubleArray {
return
this.asDynamic().slice()\n}\n\n/**
 * Returns new array which is a copy of the original array.\n * \n *
@sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\npublic actual fun BooleanArray.copyOfOf():
BooleanArray {
return withType("BooleanArray", this.asDynamic().slice())\n}\n\n/**
 * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\npublic actual fun CharArray.copyOfOf(): CharArray {
return withType("CharArray",
this.asDynamic().slice())\n}\n\n/**
 * Returns new array which is a copy of
the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero
values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
[newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
*\npublic actual fun ByteArray.copyOfOf(newSize: Int): ByteArray {
require(newSize >= 0) { "Invalid new array
size: $newSize." }\n return fillFrom(this, ByteArray(newSize))\n}\n\n/**
 * Returns new array which is a copy

```



of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size

of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun

ShortArray.copyOf(newSize: Int): ShortArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, ShortArray(newSize))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun IntArray.copyOf(newSize: Int): IntArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, IntArray(newSize))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun LongArray.copyOf(newSize: Int): LongArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"LongArray\", arrayCopyResize(this, newSize, 0L))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun FloatArray.copyOf(newSize: Int): FloatArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, FloatArray(newSize))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with zero values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return fillFrom(this, DoubleArray(newSize))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with `false` values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"BooleanArray\", arrayCopyResize(this, newSize, false))\n}\n\n/\*\*\n \* Returns new array which is a copy of the original array, resized to the given [newSize].\n \* The copy is either truncated or padded at the end with null char ( `\\u0000` ) values if necessary.\n \* \n \* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char ( `\\u0000` ) values.\n \* \n \* @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n \* \n public actual fun CharArray.copyOf(newSize: Int): CharArray {\n require(newSize >= 0) { \"Invalid new array size: \$newSize.\" }\n return withType(\"CharArray\", fillFrom(this, CharArray(newSize)))\n}\n\n/\*\*\n \* Returns new array which is



```

copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\npublic
actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\npublic
actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType("<code>BooleanArray</code>",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\npublic actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType("<code>CharArray</code>",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the specified
[element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param
toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic
actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int =
size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic
actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of
the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this
array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic
actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic
actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range

```

(inclusive) to fill, 0 by default.\n \* @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.fill(element: Long, fromIndex:
 Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n
 nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element]
value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end
of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
(exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.fill(element:
 Char, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex,
 size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n * \n *\n@Suppress("ACTUAL_WITHOUT_EXPECT",
"NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<out T>.plus(element: T): Array<T> {\n
 return this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n * \n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun ByteArray.plus(element: Byte): ByteArray {\n return plus(byteArrayOf(element))\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(element: Short):
ShortArray {\n return plus(shortArrayOf(element))\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(element: Int):

```

IntArray { \n return plus(intArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then the given [element]. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun LongArray.plus(element: Long): LongArray { \n return plus(longArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then the given [element]. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun FloatArray.plus(element: Float): FloatArray { \n return plus(floatArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then the given [element]. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun DoubleArray.plus(element: Double): DoubleArray { \n return plus(doubleArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then the given [element]. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray { \n return plus(booleanArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then the given [element]. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun CharArray.plus(element: Char): CharArray { \n return plus(charArrayOf(element)) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n @Suppress("ACTUAL\_WITHOUT\_EXPECT") \n public actual operator fun <T> Array<out T>.plus(elements: Collection<T>): Array<T> { \n return arrayPlusCollection(this, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun ShortArray.plus(elements: Collection<Short>): ShortArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun IntArray.plus(elements: Collection<Int>): IntArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun LongArray.plus(elements: Collection<Long>): LongArray { \n return arrayPlusCollection(this, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun FloatArray.plus(elements: Collection<Float>): FloatArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray { \n return arrayPlusCollection(this, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection. \n \* \n public actual operator fun CharArray.plus(elements: Collection<Char>): CharArray { \n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array. \n \* \n @Suppress("ACTUAL\_WITHOUT\_EXPECT", "NOTHING\_TO\_INLINE") \n public actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> { \n return this.asDynamic().concat(elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray { \n return primitiveArrayConcat(this, elements) \n } \n \n /\*\* \n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array. \n \* \n @Suppress("NOTHING\_TO\_INLINE") \n public actual inline operator fun ShortArray.plus(elements:

ShortArray): ShortArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n

\*\n@Suppress("NOTHING\_TO\_INLINE")\npublic actual inline operator fun CharArray.plus(elements: CharArray): CharArray { \n return primitiveArrayConcat(this, elements)\n}\n\n/\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n

\*\n@Suppress("ACTUAL\_WITHOUT\_EXPECT", "NOTHING\_TO\_INLINE")\npublic actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> { \n return this.asDynamic().concat(arrayOf(element))\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun IntArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\npublic actual fun LongArray.sort(): Unit { \n @Suppress("DEPRECATION")\n if (size > 1) sort { a: Long, b: Long -> a.compareTo(b) }\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun ByteArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun ShortArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun DoubleArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun FloatArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArray\n

\*\n@library("primitiveArraySort")\npublic actual fun CharArray.sort(): Unit { \n definedExternally\n}\n\n/\*\*\n \* Sorts the array in-place according to the natural order of its elements.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n

\*\npublic actual fun <T : Comparable<T>> Array<out T>.sort(): Unit { \n if (size > 1) sortArray(this)\n}\n\n/\*\*\n \* Sorts the array in-place according to the order specified by the given [comparison] function.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n

\*\n@Deprecated("Use sortWith instead", ReplaceWith("this.sortWith(Comparator(comparison))"))\n@DeprecatedSinceKotlin(warningSince = "1.6")\npublic fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit { \n if (size > 1) sortArrayWith(this, comparison)\n}\n\n/\*\*\n \* Sorts a range in the array in-place.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* @param

fromIndex the start of the range (inclusive) to sort, 0 by default.\n \* @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \* \n \* @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n

```

*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArrayWith(this, fromIndex, toIndex,
naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<IntArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n sortArrayWith(this.unsafeCast<Array<Long>>(), fromIndex, toIndex,
naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than

```

```

the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<FloatArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array
in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the
end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<DoubleArray>()\n subarray.sort()\n}\n\n/**\n * Sorts
a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort,
size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()\n subarray.sort()\n}\n\n/**\n * Sorts the
array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated("Use other
sorting functions from the Standard Library")\n@DeprecatedSinceKotlin(warningSince =
"1.6")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) ->
Int): Unit
{\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given
[comparison] function.\n */\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n *
Sorts the array in-place according to the order specified by the given [comparison] function.\n
*\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the
array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated("Use other
sorting functions from the Standard Library")\n@DeprecatedSinceKotlin(warningSince
= "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) -
> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by
the given [comparison] function.\n */\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n *
Sorts the array in-place according to the order specified by the given [comparison] function.\n
*\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {\n

```



```

nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order
 * specified by the given [comparison] function.\n *^\n@Deprecated("\nUse other sorting functions from the Standard
 * Library\n")\n@DeprecatedSinceKotlin(warningSince = `1.6`)\n@kotlin.internal.InlineOnly\npublic inline fun
 * CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n *
 * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It
 * means that equal elements preserve their order relative to each other after sorting.\n *^\npublic actual fun <T>
 * Array<out T>.sortWith(comparator: Comparator<in T>): Unit {\n if (size > 1) sortArrayWith(this,
 * comparator)\n}\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The sort is
 * _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param
 * fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range
 * (exclusive)
 * to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
 * or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
 * than [toIndex].\n
 *^\n@SinceKotlin(`1.4`)\n@Suppress("\nACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\n")\npublic
 * actual fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size):
 * Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArrayWith(this, fromIndex, toIndex,
 * comparator)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
 *^\npublic actual fun ByteArray.toTypedArray(): Array<Byte> {\n return js("\n[]\n").slice.call(this)\n}\n\n/**\n *
 * Returns a *typed* object array containing all of the elements of this primitive array.\n *^\npublic actual fun
 * ShortArray.toTypedArray(): Array<Short> {\n return js("\n[]\n").slice.call(this)\n}\n\n/**\n *
 * Returns a *typed* object array containing all of the elements of this primitive array.\n *^\npublic actual fun
 * IntArray.toTypedArray(): Array<Int> {\n return js("\n[]\n").slice.call(this)\n}\n\n/**\n * Returns a *typed* object
 * array containing all of the elements of this primitive array.\n *^\npublic actual fun LongArray.toTypedArray():
 * Array<Long> {\n return js("\n[]\n").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the
 * elements of this primitive array.\n *^\npublic actual fun FloatArray.toTypedArray(): Array<Float> {\n return
 * js("\n[]\n").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
 * array.\n *^\npublic actual fun DoubleArray.toTypedArray(): Array<Double> {\n return
 * js("\n[]\n").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
 * array.\n *^\npublic actual fun BooleanArray.toTypedArray(): Array<Boolean> {\n return
 * js("\n[]\n").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
 * array.\n *^\npublic actual fun CharArray.toTypedArray(): Array<Char> {\n return Array(size) { index ->
 * this[index] }\n}\n\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.\n
 *^\n@file:kotlin.jvm.JvmName("\nComparisonsKt\n")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
 * kotlin.comparisons\n\n/**\n * Compares two values using the specified functions [selectors] to calculate the result
 * of the comparison.\n * The functions are called sequentially, receive the given values [a] and [b] and return
 * [Comparable]\n * objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do
 * not\n * compare as equal, the result of that comparison is returned.\n * \n * @sample
 * samples.comparisons.Comparisons.compareValuesByWithSelectors\n
 ^\npublic fun <T> compareValuesBy(a: T, b: T, vararg selectors: (T) -> Comparable<>?): Int {\n
 * require(selectors.size > 0)\n return compareValuesByImpl(a, b, selectors)\n}\n\nprivate fun <T>
 * compareValuesByImpl(a: T, b: T, selectors: Array<out (T) -> Comparable<*>?): Int {\n for (fn in selectors) {\n
 * val v1 = fn(a)\n val v2 = fn(b)\n val diff = compareValues(v1, v2)\n if (diff != 0) return diff\n }\n
 * return 0\n}\n\n/**\n * Compares two values using the specified [selector] function to calculate the result of the
 * comparison.\n * The function is applied to the given values [a] and [b] and return [Comparable] objects.\n * The
 * result of comparison of these [Comparable] instances is returned.\n * \n * @sample

```

```

samples.comparisons.Comparisons.compareValuesByWithSingleSelector\n *\n@kotlin.internal.InlineOnly\npublic
inline fun <T> compareValuesBy(a: T, b: T, selector: (T) -> Comparable<*>?): Int {\n
 return compareValues(selector(a), selector(b))\n}\n\n/**\n * Compares two values using the specified [selector]
function to calculate the result of the comparison.\n * The function is applied to the given values [a] and [b] and
return objects of type K which are then being\n * compared with the given [comparator].\n *\n * @sample
samples.comparisons.Comparisons.compareValuesByWithComparator\n *\n@kotlin.internal.InlineOnly\npublic
inline fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in K>, selector: (T) -> K): Int {\n return
comparator.compare(selector(a), selector(b))\n}\n\n//// Not so useful without type inference for receiver of
expression\n//// compareValuesWith(v1, v2, compareBy { it.prop1 } thenByDescending { it.prop2 })\n////**\n// *
Compares two values using the specified [comparator].\n\n *\n * @Suppress("NOTHING_TO_INLINE")\n\n//public
inline fun <T> compareValuesWith(a: T, b: T, comparator: Comparator<T>): Int = comparator.compare(a,
b)\n\n/\n\n/**\n * Compares
two nullable [Comparable] values. Null is considered less than any value.\n *\n * @sample
samples.comparisons.Comparisons.compareValues\n *\npublic fun <T : Comparable<*>> compareValues(a: T?, b:
T?): Int {\n if (a === b) return 0\n if (a == null) return -1\n if (b == null) return 1\n\n @Suppress("UNCHECKED_CAST")\n return (a as Comparable<Any>).compareTo(b)\n}\n\n/**\n * Creates a
comparator using the sequence of functions to calculate a result of comparison.\n * The functions are called
sequentially, receive the given values `a` and `b` and return [Comparable]\n * objects. As soon as the [Comparable]
instances returned by a function for `a` and `b` values do not\n * compare as equal, the result of that comparison is
returned from the [Comparator].\n *\n * @sample samples.comparisons.Comparisons.compareByWithSelectors\n
\npublic fun <T> compareBy(vararg selectors: (T) -> Comparable<>?): Comparator<T> {\n
 require(selectors.size > 0)\n return Comparator
 { a, b -> compareValuesByImpl(a, b, selectors) }\n}\n\n/**\n * Creates a comparator using the function to
transform value to a [Comparable] instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.compareByWithSingleSelector\n *\n@kotlin.internal.InlineOnly\npublic inline
fun <T> compareBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n Comparator { a, b ->
compareValuesBy(a, b, selector) }\n\n/**\n * Creates a comparator using the [selector] function to transform values
being compared and then applying\n * the specified [comparator] to compare transformed values.\n *\n * @sample
samples.comparisons.Comparisons.compareByWithComparator\n *\n@kotlin.internal.InlineOnly\npublic inline
fun <T, K> compareBy(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n Comparator
{ a, b -> compareValuesBy(a, b, comparator, selector) }\n\n/**\n * Creates a descending comparator
using the function to transform value to a [Comparable]
instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.compareByDescendingWithSingleSelector\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareByDescending(crossinline selector: (T) ->
Comparable<*>?): Comparator<T> =\n Comparator { a, b -> compareValuesBy(b, a, selector) }\n\n/**\n *
Creates a descending comparator using the [selector] function to transform values being compared and then
applying\n * the specified [comparator] to compare transformed values.\n *\n * Note that an order of [comparator] is
reversed by this wrapper.\n *\n * @sample
samples.comparisons.Comparisons.compareByDescendingWithComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareByDescending(comparator: Comparator<in K>,
crossinline selector: (T) -> K): Comparator<T> =\n Comparator { a, b -> compareValuesBy(b, a, comparator,
selector) }\n\n/**\n * Creates a comparator comparing values after the primary comparator defined them equal. It
uses\n * the function
to transform value to a [Comparable] instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.thenBy\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n Comparator { a, b -
>\n val previousCompare = this@thenBy.compare(a, b)\n if (previousCompare != 0) previousCompare else

```

```

compareValuesBy(a, b, selector)\n } \n\n/**\n * Creates a comparator comparing values after the primary
comparator defined them equal. It uses\n * the [selector] function to transform values and then compares them with
the given [comparator].\n *\n * @sample samples.comparisons.Comparisons.thenByWithComparator\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> Comparator<T>.thenBy(comparator: Comparator<in K>,\n
crossinline selector: (T) -> K): Comparator<T> =\n Comparator { a, b ->\n val previousCompare =
this@thenBy.compare(a, b)\n if (previousCompare != 0) previousCompare
 else compareValuesBy(a, b, comparator, selector)\n } \n\n/**\n * Creates a descending comparator using the
primary comparator and\n * the function to transform value to a [Comparable] instance for comparison.\n *\n *
@sample samples.comparisons.Comparisons.thenByDescending\n */\n@kotlin.internal.InlineOnly\npublic inline
fun <T> Comparator<T>.thenByDescending(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n
 Comparator { a, b ->\n val previousCompare = this@thenByDescending.compare(a, b)\n if
(previousCompare != 0) previousCompare else compareValuesBy(b, a, selector)\n } \n\n/**\n * Creates a
descending comparator comparing values after the primary comparator defined them equal. It uses\n * the [selector]
function to transform values and then compares them with the given [comparator].\n *\n * @sample
samples.comparisons.Comparisons.thenByDescendingWithComparator\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T, K> Comparator<T>.thenByDescending(comparator:
 Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n Comparator { a, b ->\n val
previousCompare = this@thenByDescending.compare(a, b)\n if (previousCompare != 0) previousCompare else
compareValuesBy(b, a, comparator, selector)\n } \n\n/**\n * Creates a comparator using the primary comparator
and function to calculate a result of comparison.\n *\n * @sample
samples.comparisons.Comparisons.thenComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =\n Comparator { a,
b ->\n val previousCompare = this@thenComparator.compare(a, b)\n if (previousCompare != 0)
previousCompare else comparison(a, b)\n } \n\n/**\n * Combines this comparator and the given [comparator] such
that the latter is applied only\n * when the former considered values equal.\n *\n * @sample
samples.comparisons.Comparisons.then\n */\npublic infix
fun <T> Comparator<T>.then(comparator: Comparator<in T>): Comparator<T> =\n Comparator { a, b ->\n
val previousCompare = this@then.compare(a, b)\n if (previousCompare != 0) previousCompare else
comparator.compare(a, b)\n } \n\n/**\n * Combines this comparator and the given [comparator] such that the latter
is applied only\n * when the former considered values equal.\n *\n * @sample
samples.comparisons.Comparisons.thenDescending\n */\npublic infix fun <T>
Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =\n Comparator<T> { a, b -
->\n val previousCompare = this@thenDescending.compare(a, b)\n if (previousCompare != 0)
previousCompare else comparator.compare(b, a)\n } \n\n// Not so useful without type inference for receiver of
expression\n\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of nullable values\n *
considering `null` value less than any other value.\n * Non-null values are compared with
the provided [comparator].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n
*/\npublic fun <T : Any> nullsFirst(comparator: Comparator<in T>): Comparator<T?> =\n Comparator { a, b ->\n
when {\n a === b -> 0\n a == null -> -1\n b == null -> 1\n else ->
comparator.compare(a, b)\n } \n\n/**\n * Provides a comparator of nullable [Comparable] values\n *
considering `null` value less than any other value.\n * Non-null values are compared according to their [natural
order][naturalOrder].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsFirst(): Comparator<T?> =
nullsFirst(naturalOrder())\n\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of
nullable values\n * considering `null` value greater than any other value.\n * Non-null values are compared with the
provided [comparator].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n */\npublic fun <T : Any>
nullsLast(comparator: Comparator<in T>): Comparator<T?> =\n Comparator { a, b ->\n when {\n a

```

```

=== b -> 0\n a == null -> 1\n b == null -> -1\n else -> comparator.compare(a, b)\n }\n}\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value greater than any\n * other value.\n * Non-null values are compared according to their [natural order][naturalOrder].\n *\n * @sample\n samples.comparisons.Comparisons.nullsFirstLastComparator\n *\n @kotlin.internal.InlineOnly\n public inline fun\n <T : Comparable<T>> nullsLast(): Comparator<T?> = nullsLast(naturalOrder())\n\n /**\n * Returns a comparator\n * that compares [Comparable] objects in natural order.\n *\n * The natural order of a `Comparable` type here means\n * the order established by its `compareTo` function.\n *\n * @sample\n samples.comparisons.Comparisons.naturalOrderComparator\n\n *\n public fun <T : Comparable<T>> naturalOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")\n (NaturalOrderComparator as Comparator<T>)\n\n /**\n * Returns a comparator that compares [Comparable] objects\n * in reversed natural order.\n *\n * The natural order of a `Comparable` type here means the order established by its\n * `compareTo` function.\n *\n * @sample\n samples.comparisons.Comparisons.nullsFirstLastWithComparator\n\n *\n public fun <T : Comparable<T>> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")\n (ReverseOrderComparator as Comparator<T>)\n\n /**\n * Returns a comparator that imposes the reverse ordering\n * of this comparator.\n *\n * @sample\n samples.comparisons.Comparisons.reversed\n\n *\n @Suppress("EXTENSION_SHADOWED_BY_MEMBER")\n public fun <T> Comparator<T>.reversed():\n Comparator<T> = when (this) {\n is ReversedComparator -> this.comparator\n NaturalOrderComparator ->\n @Suppress("UNCHECKED_CAST") (ReverseOrderComparator\n as Comparator<T>)\n ReverseOrderComparator -> @Suppress("UNCHECKED_CAST")\n (NaturalOrderComparator as Comparator<T>)\n else -> ReversedComparator(this)\n }\n\n private class\n ReversedComparator<T>(public val comparator: Comparator<T>) : Comparator<T> {\n override fun compare(a:\n T, b: T): Int = comparator.compare(b, a)\n @Suppress("VIRTUAL_MEMBER_HIDDEN")\n fun reversed():\n Comparator<T> = comparator\n }\n\n private object NaturalOrderComparator : Comparator<Comparable<Any>> {\n override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)\n\n @Suppress("VIRTUAL_MEMBER_HIDDEN")\n fun reversed(): Comparator<Comparable<Any>> =\n ReverseOrderComparator\n }\n\n private object ReverseOrderComparator : Comparator<Comparable<Any>> {\n override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)\n\n @Suppress("VIRTUAL_MEMBER_HIDDEN")\n fun reversed(): Comparator<Comparable<Any>> =\n NaturalOrderComparator\n }\n\n", "/*\n * Copyright\n * 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed\n * by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n *\n @file:kotlin.jvm.JvmMultifileClass\n @file:kotlin.jvm.JvmName("StandardKt")\n npackage kotlin\n\n nimport\n kotlin.contracts.*\n\n /**\n * An exception is thrown to indicate that a method body remains to be implemented.\n\n *\n public class NotImplementedError(message: String = "An operation is not implemented.") :\n Error(message)\n\n /**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n\n *\n @kotlin.internal.InlineOnly\n public inline fun TODO(): Nothing = throw NotImplementedError()\n\n /**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n *\n * @param reason a string\n * explaining why the implementation is missing.\n *\n @kotlin.internal.InlineOnly\n public inline fun TODO(reason:\n String): Nothing = throw NotImplementedError("An\n operation is not implemented: $reason")\n\n /**\n * Calls the specified function [block] and returns its result.\n *\n * For detailed usage information see the documentation for [scope\n * functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n\n *\n @kotlin.internal.InlineOnly\n public inline fun <R> run(block: () -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return block()\n }\n\n /**\n * Calls the specified\n * function [block] with `this` value as its receiver and returns its result.\n *\n * For detailed usage information see the\n * documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n\n *\n @kotlin.internal.InlineOnly\n public inline fun <T, R> T.run(block: T.() -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return block()\n }\n\n /**\n * Calls the specified

```

function [block] with the given [receiver] as its receiver and returns its result.

\* For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#with>).

```

* Kotlin internal: public inline fun <T, R> with(receiver: T, block: T.() -> R): R {
 contract {
 callsInPlace(block, InvocationKind.EXACTLY_ONCE)
 }
 return receiver.block()
}

```

\* Calls the specified function [block] with `this` value as its receiver and returns `this` value.

\* For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#apply>).

```

* Kotlin internal: public inline fun <T> T.apply(block: T.() -> Unit): T {
 contract {
 callsInPlace(block, InvocationKind.EXACTLY_ONCE)
 }
 block()
 return this
}

```

\* Calls the specified function [block] with `this` value as its argument and returns `this` value.

\* For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#also>).

```

* Kotlin internal: public inline fun <T> T.also(block: (T) -> Unit): T {
 contract {
 callsInPlace(block, InvocationKind.EXACTLY_ONCE)
 }
 block(this)
 return this
}

```

\* Calls the specified function [block] with `this` value as its argument and returns its result.

For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#let>).

```

* Kotlin internal: public inline fun <T, R> T.let(block: (T) -> R): R {
 contract {
 callsInPlace(block, InvocationKind.EXACTLY_ONCE)
 }
 return block(this)
}

```

\* Returns `this` value if it satisfies the given [predicate] or `null`, if it doesn't.

\* For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless>).

```

* Kotlin internal: public inline fun <T> T.takeIf(predicate: (T) -> Boolean): T? {
 contract {
 callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)
 }
 return if (predicate(this)) this else null
}

```

\* Returns `this` value if it does not satisfy the given [predicate] or `null`, if it does.

\* For detailed usage information see the documentation for [scope functions](<https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless>).

```

* Kotlin internal: public inline fun <T> T.takeUnless(predicate: (T) -> Boolean): T? {
 contract {
 callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)
 }
 return if (!predicate(this)) this else null
}

```

\* Executes the given function [action] specified number of [times].

\* A zero-based index of current iteration is passed as a parameter to [action].

```

@sample samples.misc.ControlFlow.repeat
* Kotlin internal: public inline fun repeat(times: Int, action: (Int) -> Unit) {
 contract {
 callsInPlace(action)
 }
 for (index in 0 until times) {
 action(index)
 }
}

```

\* Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

\* Package kotlin.comparisons

NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt

See: <https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib>

```

kotlin.js

```

\* Returns the greater of two values.

\* If values are equal, returns the first one.

```

* Kotlin internal: public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
 return if (a >= b) a else b
}

```

\* Returns the greater of two values.

```

* Kotlin internal: public actual inline fun maxOf(a: Byte, b: Byte): Byte {
 return maxOf(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

```

\* Returns the greater of two values.

```

* Kotlin internal: public actual inline fun maxOf(a: Short, b: Short): Short {
 return maxOf(a.toInt(), b.toInt()).unsafeCast<Short>()
}

```

\* Returns the greater of two values.

```

* Kotlin internal: public actual inline fun maxOf(a: Int, b: Int): Int {
 return JsMath.max(a, b)
}

```

\* Returns the greater of two values.

```

* Kotlin internal: public actual inline fun maxOf(a: Long, b: Long): Long {
 return if (a >= b) a else b
}

```

\* Returns the greater of two values.

\* If either value

```

is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun
maxOf(a: Float, b: Float): Float {
 return JsMath.max(a, b)
}
 */
 * Returns the
greater of two values.
 */
 * If either value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Double, b: Double):
Double {
 return JsMath.max(a, b)
}
 */
 * Returns the greater of three values.
 */
 * If there are multiple
equal maximal values, returns the first of them.
 */
@SinceKotlin("1.1")
public actual fun <T :
Comparable<T>> maxOf(a: T, b: T, c: T): T {
 return maxOf(a, maxOf(b, c))
}
 */
 * Returns the greater
of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Byte,
b: Byte, c: Byte): Byte {
 return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()
}
 */
 *
Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline
fun maxOf(a: Short, b: Short, c: Short): Short {
 return JsMath.max(a.toInt(), b.toInt(),
c.toInt()).unsafeCast<Short>()
}
 */
 *
Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual
inline fun maxOf(a: Int, b: Int, c: Int): Int {
 return JsMath.max(a, b, c)
}
 */
 * Returns the greater of three
values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Long, b: Long,
c: Long): Long {
 return maxOf(a, maxOf(b, c))
}
 */
 * Returns the greater of three values.
 */
 * If any
value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun
maxOf(a: Float, b: Float, c: Float): Float {
 return JsMath.max(a, b, c)
}
 */
 * Returns the greater of three
values.
 */
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun maxOf(a: Double, b: Double, c:
Double): Double {
 return JsMath.max(a, b, c)
}
 */
 * Returns the greater of the given values.
 */
 * If
there are
multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public actual fun <T :
Comparable<T>> maxOf(a: T, vararg other: T): T {
 var max = a
 for (e in other) max = maxOf(max, e)
 return max
}
 */
 * Returns the greater of the given values.
 */
@SinceKotlin("1.4")
public actual fun
maxOf(a: Byte, vararg other: Byte): Byte {
 var max = a
 for (e in other) max = maxOf(max, e)
 return
max
}
 */
 * Returns the greater of the given values.
 */
@SinceKotlin("1.4")
public actual fun maxOf(a:
Short, vararg other: Short): Short {
 var max = a
 for (e in other) max = maxOf(max, e)
 return
max
}
 */
 * Returns the greater of the given values.
 */
@SinceKotlin("1.4")
public actual fun maxOf(a:
Int, vararg other: Int): Int {
 var max = a
 for (e in other) max = maxOf(max, e)
 return max
}
 */
 *
Returns the greater of the given values.
 */
@SinceKotlin("1.4")
public actual fun maxOf(a:
Long, vararg other: Long): Long {
 var max = a
 for (e in other) max = maxOf(max, e)
 return
max
}
 */
 * Returns the greater of the given values.
 */
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.4")
public actual fun maxOf(a: Float, vararg other: Float): Float {
 var max = a
 for (e in other) max = maxOf(max, e)
 return max
}
 */
 * Returns the greater of the given values.
 */
 * If any
value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.4")
public actual fun maxOf(a: Double, vararg other:
Double): Double {
 var max = a
 for (e in other) max = maxOf(max, e)
 return max
}
 */
 * Returns
the smaller of two values.
 */
 * If values are equal, returns the first one.
 */
@SinceKotlin("1.1")
public
actual fun <T : Comparable<T>> minOf(a: T, b: T): T {
 return if (a <= b) a else b
}
 */
 * Returns the
smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual
inline fun minOf(a: Byte, b: Byte): Byte {
 return minOf(a.toInt(), b.toInt()).unsafeCast<Byte>()
}
 */
 *
Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline
fun minOf(a: Short, b: Short): Short {
 return minOf(a.toInt(), b.toInt()).unsafeCast<Short>()
}
 */
 *
Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline
fun minOf(a: Int, b: Int): Int {
 return JsMath.min(a, b)
}
 */
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("NOTHING_TO_INLINE")
public actual inline fun minOf(a: Long, b:
Long): Long {
 return if (a <= b) a else b
}
 */
 * Returns the smaller of two values.
 */
 * If either value
is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public actual inline fun

```

```

minOf(a: Float, b: Float): Float {\n return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller
of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double):
Double {\n return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If there are multiple
equal minimal values, returns the first of them.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual fun <T :
Comparable<T>> minOf(a: T, b: T, c: T): T {\n return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of
three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Byte, b:
Byte, c: Byte): Byte {\n return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns
the smaller of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
minOf(a: Short, b: Short, c: Short): Short {\n return JsMath.min(a.toInt(), b.toInt(),
c.toInt()).unsafeCast<Short>()\n}\n\n/**\n *
Returns the smaller of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline
fun minOf(a: Int, b: Int, c: Int): Int {\n return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three
values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Long, b: Long,
c: Long): Long {\n return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
minOf(a: Float, b: Float, c: Float): Float {\n return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three
values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double, c:
Double): Double {\n return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If
there are multiple
equal minimal values, returns the first of them.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic actual fun <T :
Comparable<T>> minOf(a: T, vararg other: T): T {\n var min = a\n for (e in other) min = minOf(min, e)\n return
min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
actual fun minOf(a: Byte, vararg other: Byte): Byte {\n var min = a\n for (e in other) min = minOf(min, e)\n return
min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
actual fun minOf(a: Short, vararg other: Short): Short {\n var min = a\n for (e in other) min = minOf(min, e)\n return
min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
actual fun minOf(a: Int, vararg other: Int): Int {\n var min = a\n for (e in other) min = minOf(min, e)\n return
min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
actual fun minOf(a: Long, vararg other: Long): Long {\n var min = a\n for (e in other) min = minOf(min, e)\n return
min\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic actual fun minOf(a: Float, vararg other: Float): Float {\n
 var min = a\n for (e in other) min = minOf(min, e)\n return min\n}\n\n/**\n * Returns the smaller of the given
values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic actual fun minOf(a: Double, vararg other:
Double): Double {\n var min = a\n for (e in other) min = minOf(min, e)\n return min\n}\n\n"/>\n\n/* Copyright
2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n@n\// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic
value class ULong @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi
internal val data: Long) : Comparable<ULong> {\n\n companion object {\n\n /**\n * A constant holding
the minimum value an instance of ULong can have.\n */\n public const val MIN_VALUE: ULong =
ULong(0)\n\n /**\n * A constant holding the maximum value an instance of ULong can have.\n */\n
 public const val MAX_VALUE: ULong = ULong(-1)\n\n /**\n * The number of bytes used to represent
an instance of ULong in a binary form.\n */\n public const val SIZE_BYTES: Int = 8\n\n /**\n
 * The number of bits used to represent an instance of ULong in a binary form.\n */\n public const val

```





```

 @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): ULong =
this.rem(other.toULong())\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())\n /**\n * Calculates the remainder of
truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n
 @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong): ULong = ulongRemainder(this,
other)\n\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative
infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n
 */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UByte): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division
and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UShort): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division
and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
UInt): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division
and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
ULong): ULong =
div(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n *
The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and
truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte):
UByte = this.mod(other.toULong()).toUByte()\n /**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types,
the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
 public inline fun mod(other: UShort): UShort = this.mod(other.toULong()).toUShort()\n /**\n * Calculates the
remainder of flooring division of this value
by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
 public inline fun mod(other: UInt): UInt = this.mod(other.toULong()).toUInt()\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
 */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = rem(other)\n\n /**\n *
Returns this value incremented by one.\n * \n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n
 public inline operator fun inc(): ULong = ULong(data.inc())\n\n /**\n * Returns this value decremented by one.\n
 * \n * @sample samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
dec(): ULong = ULong(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. *\n */\n
 @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this,
other)\n\n /**\n * Creates a range from this value up to but excluding the specified [other] value.\n * \n *
If the [other] value is less than or equal to `this` value, then the returned range is empty.\n */\n @SinceKotlin("1.7")\n
 @ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: ULong):
ULongRange = this.until
other)\n\n /**\n * Shifts this value left by the [bitCount] number of bits.\n * \n * Note that only the six
lowest-order bits of the [bitCount] are used as the shift distance.\n * \n * The shift distance actually used is
therefore always in the range `0..63`.\n */\n @kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount:
Int): ULong = ULong(data shl bitCount)\n\n /**\n * Shifts this value right by the [bitCount] number of bits,
filling the leftmost bits with zeros.\n * \n * Note that only the six lowest-order bits of the [bitCount] are
used as the shift distance.\n * \n * The shift distance

```

```

actually used is therefore always in the range `0..63`.
 @kotlin.internal.InlineOnly
 public inline infix fun shr(bitCount: Int): ULong = ULong(data ushr bitCount)
 /** Performs a bitwise AND operation between the two values.
 @kotlin.internal.InlineOnly
 public inline infix fun and(other: ULong): ULong = ULong(this.data and other.data)
 /** Performs a bitwise OR operation between the two values.
 @kotlin.internal.InlineOnly
 public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)
 /** Performs a bitwise XOR operation between the two values.
 @kotlin.internal.InlineOnly
 public inline infix fun xor(other: ULong): ULong = ULong(this.data xor other.data)
 /** Inverts the bits in this value.
 @kotlin.internal.InlineOnly
 public inline fun inv(): ULong = ULong(data.inv())
 /** Converts this [ULong] value to [Byte].
 * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents the same numerical value as this `ULong`.
 * The resulting `Byte` value is represented by the least significant 8 bits of this `ULong` value.
 * Note that the resulting `Byte` value may be negative.
 @kotlin.internal.InlineOnly
 public inline fun toByte(): Byte = data.toByte()
 /** Converts this [ULong] value to [Short].
 * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents the same numerical value as this `ULong`.
 * The resulting `Short` value is represented by the least significant 16 bits of this `ULong` value.
 * Note that the resulting `Short` value may be negative.
 @kotlin.internal.InlineOnly
 public inline fun toShort(): Short = data.toShort()
 /** Converts this [ULong] value to [Int].
 * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int` value represents the same numerical value as this `ULong`.
 * The resulting `Int` value is represented by the least significant 32 bits of this `ULong` value.
 * Note that the resulting `Int` value may be negative.
 @kotlin.internal.InlineOnly
 public inline fun toInt(): Int = data.toInt()
 /** Converts this [ULong] value to [Long].
 * If this value is less than or equals to [Long.MAX_VALUE], the resulting `Long` value represents the same numerical value as this `ULong`. Otherwise the result is negative.
 * The resulting `Long` value has the same binary representation as this `ULong` value.
 @kotlin.internal.InlineOnly
 public inline fun toLong(): Long = data
 /** Converts this [ULong] value to [UByte].
 * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `ULong`.
 * The resulting `UByte` value is represented by the least significant 8 bits of this `ULong` value.
 @kotlin.internal.InlineOnly
 public inline fun toUByte(): UByte = data.toUByte()
 /** Converts this [ULong] value to [UShort].
 * If this value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same numerical value as this `ULong`.
 * The resulting `UShort` value is represented by the least significant 16 bits of this `ULong` value.
 @kotlin.internal.InlineOnly
 public inline fun toUShort(): UShort = data.toUShort()
 /** Converts this [ULong] value to [UInt].
 * If this value is less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents the same numerical value as this `ULong`.
 * The resulting `UInt` value is represented by the least significant 32 bits of this `ULong` value.
 @kotlin.internal.InlineOnly
 public inline fun toUInt(): UInt = data.toUInt()
 /** Returns this value.
 @kotlin.internal.InlineOnly
 public inline fun toULong(): ULong = this
 /** Converts this [ULong] value to [Float].
 * The resulting value is the closest `Float` to this `ULong` value.
 * In case when this `ULong` value is exactly between two `Float`s, the one with zero at least significant bit of mantissa is selected.
 @kotlin.internal.InlineOnly
 public inline fun toFloat(): Float = this.toDouble().toFloat()
 /** Converts this [ULong] value to [Double].
 * The resulting value is the closest `Double` to this `ULong` value.
 * In case when this `ULong` value is exactly between two `Double`s, the one with zero at least significant bit of mantissa is selected.
 @kotlin.internal.InlineOnly
 public inline fun toDouble(): Double = ulongToDouble(data)
 public override fun toString(): String = ulongToString(data)
 /** Converts this [Byte] value to [ULong].
 * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Byte`.
 * The least significant 8 bits of the resulting `ULong` value are the same as the

```

bits of this `Byte` value,\n \* whereas the most significant 56 bits are filled with the sign bit of this value.\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this
```

[Short] value to [ULong].\n \* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Short`.\n \* The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short` value,\n \* whereas the most significant 48 bits are filled with the sign bit of this value.\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this [Int] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Int`.\n *\n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value,\n *\n * whereas the most significant 32 bits are filled with the sign bit of this value.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline
```

```
fun Int.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this [Long] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`.\n *\n * The resulting `ULong` value has the same binary representation as this `Long` value.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toULong(): ULong = ULong(this)\n/**\n * Converts this [Float] value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())\n/**\n * Converts this [Double] value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Double` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n
```

```
\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Double.toULong(): ULong = doubleToULong(this)\n","/\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
```

```
\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1.\n *\n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> List<T>.component1(): T {\n * return get(0)\n * }\n **\n * Returns 2nd *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 2.\n *\n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> List<T>.component2(): T {\n * return get(1)\n * }\n **\n * Returns 3rd *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 3.\n *\n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> List<T>.component3(): T {\n * return get(2)\n * }\n **\n * Returns 4th *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.\n *\n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> List<T>.component4(): T {\n * return get(3)\n * }\n **\n * Returns 5th *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 5.\n *\n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> List<T>.component5(): T {\n * return get(4)\n * }\n **\n * Returns `true` if [element] is found in the collection.\n *\n * @kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {\n * if (this is Collection)\n * return contains(element)\n * return indexOf(element) >= 0\n * }\n **\n * Returns an element at the given [index] or
```

```

throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this collection.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n * \n\npublic fun <T> Iterable<T>.elementAt(index: Int): T {\n
if (this is List)\n return get(index)\n return elementAtOrElse(index) { throw
IndexOutOfBoundsException("Collection doesn't contain element at index $index.") }\n}\n\n/**\n * Returns an
element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this list.\n
* \n * @sample
samples.collections.Collections.Elements.elementAt\n * \n\n@kotlin.internal.InlineOnly\npublic inline fun <T>
List<T>.elementAt(index: Int): T {\n return get(index)\n}\n\n/**\n * Returns an element at the given [index] or
the result of calling the [defaultValue] function if the [index] is out of bounds of this collection.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n * \n\npublic fun <T>
Iterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n if (this is List)\n return
this.getOrElse(index, defaultValue)\n if (index < 0)\n return defaultValue(index)\n val iterator = iterator()\n var count = 0\n while (iterator.hasNext()) {\n val element = iterator.next()\n if (index == count++)\n return element\n }\n return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n * \n\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) ->
T): T {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or `null` if the [index] is out of bounds of this collection.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n * \n\npublic fun <T>
Iterable<T>.elementAtOrNull(index: Int): T? {\n if (this is List)\n return this.getOrElse(index)\n if (index <
0)\n return null\n val iterator = iterator()\n var count = 0\n while (iterator.hasNext()) {\n val element =
iterator.next()\n if (index == count++)\n return element\n }\n return null\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this list.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n * \n\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAtOrNull(index: Int): T? {\n return
this.getOrElse(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {\n return
firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such
element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.findLast(predicate: (T) -> Boolean): T? {\n
return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no
such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\npublic
inline fun <T> List<T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the collection is empty.\n * \n\npublic fun
<T> Iterable<T>.first(): T {\n when (this) {\n is List -> return this.first()\n else -> {\n val iterator =
iterator()\n if (!iterator.hasNext())\n throw NoSuchElementException("Collection is empty.")\n return iterator.next()\n }\n }\n}\n\n/**\n * Returns the first element.\n * \n * @throws
NoSuchElementException if the list is empty.\n * \n\npublic fun <T> List<T>.first(): T {\n if (isEmpty())\n throw NoSuchElementException("List is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element
matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n * \n\npublic
inline fun <T> Iterable<T>.first(predicate: (T) -> Boolean):
T {\n for (element in this) if (predicate(element)) return element\n throw
NoSuchElementException("Collection contains no element matching the predicate.")\n}\n\n/**\n * Returns the
first non-null value produced by [transform] function being applied to elements of this collection in iteration order,\n
* or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.firstNotNullOf
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the collection was transformed to a non-null value.")\n}\n\n/**\n *
Returns the first non-null value produced by [transform] function being applied to elements of this collection in
iteration order.\n * or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this) {\n val result =
transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n *
Returns the first element, or `null` if the collection is empty.\n *^@public fun <T> Iterable<T>.firstOrNull(): T? {\n
when (this) {\n is List -> {\n if (isEmpty())\n return null\n else\n return this[0]\n }\n else -> {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n return iterator.next()\n }\n}\n}\n\n/**\n * Returns the first element, or `null` if the list is empty.\n *^@public
fun <T> List<T>.firstOrNull(): T? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns
the first element matching the given [predicate], or `null` if element was not found.\n *^@public inline fun <T>
Iterable<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return
element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this list.\n *^@kotlin.internal.InlineOnly\npublic inline
fun <T> List<T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this list.\n * \n * @sample samples.collections.Collections.Elements.getOrElse\n *^@public fun <T> List<T>.getOrElse(index: Int): T? {\n return if (index >= 0 && index <= lastIndex) get(index)
else null\n}\n\n/**\n * Returns first index of [element], or -1 if the collection does
not contain element.\n *^@public fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.indexOf(element: T): Int
{\n if (this is List) return this.indexOf(element)\n var index = 0\n for (item in this) {\n
checkIndexOverflow(index)\n if (element == item)\n return index\n index++\n }\n return -
1\n}\n\n/**\n * Returns first index of [element], or -1 if the list does not contain element.\n *^@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases\npublic fun <@kotlin.internal.OnlyInputTypes T> List<T>.indexOf(element: T): Int {\n return
indexOf(element)\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
collection does not contain such element.\n *^@public inline fun <T> Iterable<T>.indexOfFirst(predicate: (T) ->
Boolean): Int {\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if
(predicate(item))\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the collection does not contain such element.\n *^@public inline fun
<T> Iterable<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n var lastIndex = -1\n var index = 0\n for (item
in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n lastIndex = index\n index++\n }\n return lastIndex\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the list
does not contain such element.\n *^@public inline fun <T> List<T>.indexOfLast(predicate:
(T) -> Boolean): Int {\n val iterator = this.listIterator(size)\n while (iterator.hasPrevious()) {\n if
(predicate(iterator.previous())) {\n return iterator.nextIndex()\n }\n }\n return -1\n}\n\n/**\n *
Returns the last element.\n * \n * @throws NoSuchElementException if the collection is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^@public fun <T> Iterable<T>.last(): T {\n when (this) {\n is
List -> return this.last()\n else -> {\n val iterator = iterator()\n if (!iterator.hasNext())\n

```

```

throw NoSuchElementException("Collection is empty.")\n var last = iterator.next()\n while
(iterator.hasNext())\n last = iterator.next()\n return last\n }\n }\n\n/**\n * Returns the last
element.\n * \n * @throws NoSuchElementException if the list is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic fun <T> List<T>.last(): T {\n if (isEmpty())\n throw NoSuchElementException("List is
empty.")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n *
@throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic inline fun <T> Iterable<T>.last(predicate: (T) ->
Boolean): T {\n var last: T? = null\n var found = false\n for (element in this) {\n if (predicate(element))
{\n last = element\n found = true\n }\n }\n if (!found) throw
NoSuchElementException("Collection contains no element matching the predicate.")\n @Suppress("UNCHECKED_CAST")\n return last as T\n}\n\n/**\n * Returns the last element matching the
given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic inline fun <T> List<T>.last(predicate: (T) -> Boolean): T {\n val iterator = this.listIterator(size)\n while (iterator.hasPrevious()) {\n val element = iterator.previous()\n if (predicate(element)) return
element\n }\n throw NoSuchElementException("List contains no element matching the predicate.")\n}\n\n/**\n * Returns last index of [element], or -1 if the collection does not contain element.\n */\npublic fun
<@kotlin.internal.OnlyInputTypes T> Iterable<T>.lastIndexOf(element: T): Int {\n if (this is List) return
this.lastIndexOf(element)\n var lastIndex = -1\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (element == item)\n lastIndex = index\n index++\n }\n return
lastIndex\n}\n\n/**\n * Returns last index of [element], or -1 if the list does not contain element.\n */\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases\npublic
fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {\n return
lastIndexOf(element)\n}\n\n/**\n * Returns the last element, or `null` if the collection is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic fun <T> Iterable<T>.lastOrNull(): T? {\n when (this)
{\n is List -> return if (isEmpty()) null else this[size - 1]\n else -> {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n var last = iterator.next()\n while (iterator.hasNext())\n last = iterator.next()\n return last\n }\n }\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null`
if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\npublic inline
fun <T> Iterable<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n var last: T? = null\n for (element in this) {\n if (predicate(element)) {\n last = element\n }\n }\n return last\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*/\npublic inline fun <T> List<T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n val iterator = this.listIterator(size)\n while (iterator.hasPrevious()) {\n val element =
iterator.previous()\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns a random
element from this collection.\n * \n * @throws NoSuchElementException if this collection is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.random():
T {\n return random(Random)\n}\n\n/**\n * Returns a random element from this collection using the specified
source of randomness.\n * \n * @throws NoSuchElementException if this collection is empty.\n */\n@SinceKotlin("1.3")\npublic fun <T> Collection<T>.random(random: Random): T {\n if (isEmpty())\n throw NoSuchElementException("Collection is empty.")\n return elementAt(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this collection, or `null` if this collection is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic

```

```

c inline fun <T> Collection<T>.randomOrNull(): T? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this collection using the specified source of randomness, or `null` if this collection is empty.\n
*\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
Collection<T>.randomOrNull(random: Random):
T? {\n if (isEmpty())\n return null\n return elementAt(random.nextInt(size))\n}\n\n/**\n * Returns the
single element, or throws an exception if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.single(): T {\n when (this) {\n is List -> return this.single()\n else -> {\n val iterator =
iterator()\n if (!iterator.hasNext())\n throw NoSuchElementException("Collection is empty.")\n val single = iterator.next()\n if (iterator.hasNext())\n throw
IllegalArgumentOutOfRangeException("Collection has more than one element.")\n return single\n }\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the list is empty or has more than one
element.\n */\npublic fun <T> List<T>.single(): T {\n return when (size) {\n 0 -> throw
NoSuchElementException("List is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentOutOfRangeException("List has more than one element.")\n }\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic
inline fun <T> Iterable<T>.single(predicate: (T) -> Boolean): T {\n var single: T? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw
IllegalArgumentOutOfRangeException("Collection contains more than one matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Collection contains no element
matching the predicate.")\n @SuppressWarnings("UNCHECKED_CAST")\n return single as T\n}\n\n/**\n * Returns
single element, or `null` if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.singleOrNull(): T? {\n when (this) {\n is List -> return if (size == 1) this[0] else null\n
 else -> {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n val single
= iterator.next()\n if (iterator.hasNext())\n return null\n return single\n }\n }\n}\n\n/**\n * Returns single element, or `null` if the list is empty or has more than one element.\n */\npublic fun
<T> List<T>.singleOrNull(): T? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic
inline fun <T> Iterable<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n var single: T? = null\n var found =
false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list
containing
all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Iterable<T>.drop(n: Int):
List<T> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n if (n == 0) return toList()\n val list: ArrayList<T>\n if (this is Collection<*>) {\n val resultSize = size - n\n if (resultSize <= 0)\n return emptyList()\n if (resultSize == 1)\n return listOf(last())\n list = ArrayList<T>(resultSize)\n if (this is List<T>) {\n if (this is RandomAccess) {\n for (index in n until size)\n list.add(this[index])\n } else {\n for (item in listIterator(n))\n list.add(item)\n }\n }\n return list\n }\n else {\n list = ArrayList<T>()\n var count =
0\n for (item in this) {\n if (count >= n) list.add(item) else ++count\n }\n return
list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun <T> List<T>.dropLast(n: Int): List<T> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T>
List<T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n if (!isEmpty()) {\n val iterator =
listIterator(size)\n while (iterator.hasPrevious()) {\n if (!predicate(iterator.previous())) {\n return

```

```

take(iterator.nextIndex() + 1)\n
 }\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first
elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun <T> Iterable<T>.dropWhile(predicate:
(T) -> Boolean): List<T> {\n var yielding = false\n val list = ArrayList<T>()\n for (item in this)\n if
(yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n
 }\n return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n * \npublic inline fun <T> Iterable<T>.filter(predicate: (T)
-> Boolean): List<T> {\n return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only
elements matching the given [predicate].\n * @param [predicate] function that takes the
index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n *
@sample samples.collections.Collections.Filtering.filterIndexed\n * \npublic inline fun <T>
Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n return
filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate] to
the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n * \npublic inline fun <T, C : MutableCollection<in T>>
Iterable<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n forEachIndexed { index,
element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n *
Returns a list containing all elements
that are instances of specified type parameter R.\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n * \npublic inline fun <reified R>
Iterable<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n return
filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n * \npublic inline fun <reified R, C :
MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination: C): C {\n for (element in this) if (element is
R) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n * \npublic inline fun <T>
Iterable<T>.filterNot(predicate: (T) -> Boolean): List<T> {\n return filterNotTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list
containing all elements that are not `null`.\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNull\n * \npublic fun <T : Any> Iterable<T?>.filterNotNull():
List<T> {\n return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the
given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n * \npublic fun <C
: MutableCollection<in T>, T : Any> Iterable<T?>.filterNotNullTo(destination: C): C {\n for (element in this) if
(element != null) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n *
\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterNotTo(destination: C, predicate: (T) ->
Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return
destination\n}\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample
samples.collections.Collections.Filtering.filterTo\n * \npublic inline fun <T, C : MutableCollection<in T>>
Iterable<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if (predicate(element))
destination.add(element)\n return destination\n}\n}\n\n/**\n * Returns a list containing elements at indices in the
specified [indices] range.\n * \npublic fun <T> List<T>.slice(indices: IntRange): List<T> {\n if
(indices.isEmpty()) return listOf()\n return this.subList(indices.start, indices.endInclusive + 1).toList()\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n * \npublic fun <T> List<T>.slice(indices:

```



```

Iterable<Int>): List<T> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<T>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T>
Iterable<T>.take(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (this is Collection<T>) {\n if (n >= size) return toList()\n if (n == 1) return listOf(first())\n }\n var count = 0\n val list = ArrayList<T>(n)\n for (item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> List<T>.takeLast(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(last())\n val list = ArrayList<T>(n)\n if (this is RandomAccess) {\n for (index in size - n until size)\n list.add(this[index])\n } else {\n for (item in listIterator(size - n))\n list.add(item)\n }\n return list}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> List<T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n if (isEmpty())\n return emptyList()\n val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n if (!predicate(iterator.previous())) {\n iterator.next()\n val expectedSize = size - iterator.nextIndex()\n if (expectedSize == 0) return emptyList()\n return ArrayList<T>(expectedSize).apply {\n while (iterator.hasNext())\n add(iterator.next())\n }\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> Iterable<T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n val list = ArrayList<T>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list}\n\n/**\n * Reverses elements in the list in-place.\n */\npublic expect fun <T> MutableList<T>.reverse(): Unit\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun <T> Iterable<T>.reversed(): List<T> {\n if (this is Collection && size <= 1) return toList()\n val list = toMutableList()\n list.reverse()\n return list}\n\n/**\n * Randomly shuffles elements in this list in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.3\")\npublic fun <T> MutableList<T>.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n this[j] = this.set(i, this[j])\n }\n}\n\n/**\n * Sorts elements in the list in-place according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> MutableList<T>.sortBy(crossinline selector: (T) -> R?): Unit {\n if (size > 1)\n sortWith(compareBy(selector))\n}\n\n/**\n * Sorts elements in the list in-place descending according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> MutableList<T>.sortByDescending(crossinline selector: (T) -> R?): Unit {\n if (size > 1)\n sortWith(compareByDescending(selector))\n}\n\n/**\n * Sorts elements in the list in-place descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit {\n sortWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Iterable<T>.sorted(): List<T> {\n if (this is Collection) {\n if (size <= 1)\n return this.toList()\n @Suppress(\"UNCHECKED_CAST\")\n return (toArray<Comparable<T>>())

```

```

as Array<T>).apply {
 sort() }.asList()\n } \n return toMutableList().apply { sort() }\n}\n\n/**\n * Returns a list of all elements sorted
according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample
samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.sortedBy(crossinline selector: (T) -> R?): List<T> {\n return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural
sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal
elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {\n return
sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_.
It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T :
Comparable<T>> Iterable<T>.sortedDescending(): List<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is _stable_. It means
that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T>
Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> {\n if (this is Collection) {\n if (size <= 1)
return this.toList()\n @Suppress("UNCHECKED_CAST")\n return (toTypedArray<Any?>()) as
Array<T>).apply { sortWith(comparator) }.asList()\n } \n return toMutableList().apply { sortWith(comparator)
}\n}\n\n/**\n * Returns an array of Boolean containing all of the elements of this collection.\n */\npublic
fun Collection<Boolean>.toBooleanArray(): BooleanArray {\n val result = BooleanArray(size)\n var index =
0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Byte
containing all of the elements of this collection.\n */\npublic fun Collection<Byte>.toByteArray(): ByteArray {\n
val result = ByteArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return
result\n}\n\n/**\n * Returns an array of Char containing all of the elements of this collection.\n */\npublic fun
Collection<Char>.toCharArray(): CharArray {\n val result = CharArray(size)\n var index = 0\n for (element in
this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Double containing all of the
elements of this collection.\n */\npublic fun Collection<Double>.toDoubleArray(): DoubleArray {\n val result =
DoubleArray(size)\n var index = 0\n for (element
in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Float containing all of
the elements of this collection.\n */\npublic fun Collection<Float>.toFloatArray(): FloatArray {\n val result =
FloatArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return
result\n}\n\n/**\n * Returns an array of Int containing all of the elements of this collection.\n */\npublic fun
Collection<Int>.toIntArray(): IntArray {\n val result = IntArray(size)\n var index = 0\n for (element in this)\n
 result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Long containing all of the elements
of this collection.\n */\npublic fun Collection<Long>.toLongArray(): LongArray {\n val result =
LongArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return
result\n}\n\n/**\n * Returns an array of Short containing all of the elements of this
collection.\n */\npublic fun Collection<Short>.toShortArray(): ShortArray {\n val result = ShortArray(size)\n
var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns a
[Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given collection.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map
preserves the entry iteration order of the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.associate\n */\npublic inline fun <T, K, V>
Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n val capacity =
mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K,
V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given collection indexed
by the key\n * returned from [keySelector] function applied to

```

each element.  
 \* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
 \* The returned map preserves the entry iteration order of the original collection.  
 @sample samples.collections.Collections.Transformations.associateBy  
 \*  
 public inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {  
 val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)  
 return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)  
 }  
 \* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given collection.  
 \* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
 \* The returned map preserves the entry iteration order of the original collection.  
 @sample samples.collections.Collections.Transformations.associateByWithValueTransform  
 \*  
 public inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {  
 val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)  
 return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)  
 }  
 \* Populates and returns the [destination] mutable map with key-value pairs,  
 \* where key is provided by the [keySelector] function applied to each element of the given collection  
 \* and value is the element itself.  
 \* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
 @sample samples.collections.Collections.Transformations.associateByTo  
 \*  
 public inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {  
 for (element in this) {  
 destination.put(keySelector(element), element)  
 }  
 return destination  
 }  
 \* Populates and returns the [destination] mutable map with key-value pairs,  
 \* where key is provided by the [keySelector] function and  
 \* and value is provided by the [valueTransform] function applied to elements of the given collection.  
 \* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.  
 @sample samples.collections.Collections.Transformations.associateByToWithValueTransform  
 \*  
 public inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {  
 for (element in this) {  
 destination.put(keySelector(element), valueTransform(element))  
 }  
 return destination  
 }  
 \* Populates and returns the [destination] mutable map with key-value pairs  
 \* provided by [transform] function applied to each element of the given collection.  
 \* If any of two pairs would have the same key the last one gets added to the map.  
 @sample samples.collections.Collections.Transformations.associateTo  
 \*  
 public inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {  
 for (element in this) {  
 destination += transform(element)  
 }  
 return destination  
 }  
 \* Returns a [Map] where keys are elements from the given collection and values are  
 \* produced by the [valueSelector] function applied to each element.  
 \* If any two elements are equal, the last one gets added to the map.  
 \* The returned map preserves the entry iteration order of the original collection.  
 @sample samples.collections.Collections.Transformations.associateWith  
 \*  
 @SinceKotlin("1.3")  
 public inline fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {  
 val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))  
 return associateWithTo(result, valueSelector)  
 }  
 \* Populates and returns the [destination] mutable map with key-value pairs for each element of the given collection,  
 \* where key is the element itself and value is provided by the [valueSelector] function applied to that key.  
 \* If any two elements are equal, the last one overwrites the former value in the map.  
 @sample samples.collections.Collections.Transformations.associateWithTo  
 \*  
 @SinceKotlin("1.3")  
 public inline fun <K, V, M : MutableMap<in K, in V>> Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {  
 for (element in this) {  
 destination.put(element, valueSelector(element))  
 }  
 return destination  
 }  
 \* Appends all elements to the given [destination] collection.  
 public fun <T, C : MutableCollection<in T>> Iterable<T>.toCollection(destination: C): C {  
 for (item in this) {  
 destination.add(item)  
 }  
 return destination  
 }  
 \* Returns a new [HashSet] of all elements.

```

*^public fun <T> Iterable<T>.toHashSet():
HashSet<T> {
 return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))
}

Returns a [List] containing all elements.

*^public fun <T> Iterable<T>.toList(): List<T> {
 if (this is Collection) {
 return when (size) {
 0 -> emptyList()
 1 -> listOf(if (this is List) get(0) else
iterator().next())
 else -> this.toMutableList()
 }
 }
 return
this.toMutableList().optimizeReadOnlyList()
}

Returns a new [MutableList] filled with all elements of
this collection.

*^public fun <T> Iterable<T>.toMutableList(): MutableList<T> {
 if (this is Collection<T>)
 return this.toMutableList()
 return toCollection(ArrayList<T>())
}

Returns a new [MutableList]
filled with all elements of this collection.

*^public fun <T> Collection<T>.toMutableList(): MutableList<T> {
 return ArrayList(this)
}

Returns a [Set] of all elements.

* The
returned set preserves the element iteration order of the original collection.

*^public fun <T>
Iterable<T>.toSet(): Set<T> {
 if (this is Collection) {
 return when (size) {
 0 -> emptySet()
 1 -> setOf(if (this is List) this[0] else iterator().next())
 else ->
toCollection(LinkedHashSet<T>(mapCapacity(size)))
 }
 }
 return
toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()
}

Returns a single list of all elements
yielded from results of [transform] function being invoked on each element of original collection.

* @sample
samples.collections.Collections.Transformations.flatMap

*^public inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
 return flatMapTo(ArrayList<R>(),
transform)
}

Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original collection.

* @sample
samples.collections.Collections.Transformations.flatMap

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapSequence")
public inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Sequence<R>): List<R> {
 return flatMapTo(ArrayList<R>(),
transform)
}

Returns a single list of all elements yielded from results of [transform] function being
invoked on each element
* and its index in the original collection.

* @sample
samples.collections.Collections.Transformations.flatMapIndexed

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
@kotlin.internal.InlineOnly
public
inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {
 return
flatMapIndexedTo(ArrayList<R>(), transform)
}

Returns a single
list of all elements yielded from results of [transform] function being invoked on each element
* and its index in
the original collection.

* @sample samples.collections.Collections.Transformations.flatMapIndexed

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedSequence")
@kotlin.internal.InlineOnly
public
inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {
 return
flatMapIndexedTo(ArrayList<R>(), transform)
}

Appends all elements yielded from results of
[transform] function being invoked on each element
* and its index in the original collection, to the given
[destination].

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public
inline
fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T)
-> Iterable<R>): C {
 var index = 0
 for (element in this) {
 val list =
transform(checkIndexOverflow(index++), element)
 destination.addAll(list)
 }
 return
destination
}

Appends all elements yielded from results of [transform] function being invoked on each
element
* and its index in the original collection, to the given [destination].

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npu
blic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(checkIndexOverflow(index++), element)\n destination.addAll(list)\n
 }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being
invoked on each element of original collection, to the given [destination].\n */\npublic inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original collection, to the given [destination].\n */\n\n*\/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapSequenceTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n for
(element in this) {\n val list = transform(element)\n
 destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original collection by
the key returned by the given [keySelector] function\n * applied to each element and returns a map where each
group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration
order of the keys produced from the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K>
Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to
each element of the original collection\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The
returned map preserves the entry iteration
order of the keys produced from the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V>
Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements
of the original collection by the key returned by the given [keySelector] function\n * applied to each element and
puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <T, K, M : MutableMap<in K, MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) ->
K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<T>() }\n list.add(element)\n
 }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each
element of the original collection\n * by the key returned by the given [keySelector] function applied to the
element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n *
@return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Creates a [Grouping] source from a collection to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to
extract a key from each element.\n * \n * @sample samples.collections.Grouping.groupingByEachCount\n */\n\n*\/\n@SinceKotlin("1.1")\npublic inline fun <T, K> Iterable<T>.groupingBy(crossinline keySelector: (T) -> K):
Grouping<T, K> {\n return object : Grouping<T, K> {\n override fun sourceIterator(): Iterator<T> =
this@groupingBy.iterator()\n override fun keyOf(element: T): K = keySelector(element)\n }\n}\n\n/**\n *

```

Returns a list containing the results of applying the given [transform] function to each element in the original collection.

```

@sample samples.collections.Collections.Transformations.map
public inline fun <T, R>
Iterable<T>.map(transform: (T) -> R): List<R> {
 return mapTo(ArrayList<R>(collectionSizeOrDefault(10)),
transform)
}

```

Returns a list containing the results of applying the given [transform] function to each element and its index in the original collection.

```

@param [transform] function that takes the index of an
element and the element itself
and returns the result of the transform applied to the element.
public inline
fun <T, R> Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {
 return
mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

```

Returns a list containing only the non-null results of applying the given [transform] function to each element and its index in the original collection.

```

@param [transform] function that takes the index of an element and the element itself
and returns the result of the transform applied to the element.
public inline fun <T, R : Any>
Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {
 return
mapIndexedNotNullTo(ArrayList<R>(), transform)
}

```

Applies the given [transform] function to each element and its index in the original collection and appends only the non-null results to the given [destination].

```

@param [transform] function that takes
the index of an element and the element itself
and returns the result of the transform applied to the element.
public inline fun <T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapIndexedNotNullTo(destination:
C, transform: (index: Int, T) -> R?): C {
 forEachIndexed { index, element -> transform(index, element)?.let {
destination.add(it) } }
 return destination
}

```

Applies the given [transform] function to each element and its index in the original collection and appends the results to the given [destination].

```

@param [transform]
function that takes the index of an element and the element itself
and returns the result of the transform applied
to the element.
public inline fun <T, R, C : MutableCollection<in R>>
Iterable<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
 var index = 0
 for (item in this)
 destination.add(transform(checkIndexOverflow(index++), item))
 return destination
}

```

Returns a list containing only the non-null results of applying the given [transform] function to each element in the original collection.

```

@sample samples.collections.Collections.Transformations.mapNotNull
public inline fun <T, R : Any>
Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
 return
mapNotNullTo(ArrayList<R>(), transform)
}

```

Applies the given [transform] function to each element in the original collection and appends only the non-null results to the given [destination].

```

public inline fun
<T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
 forEach { element -> transform(element)?.let { destination.add(it) } }
 return destination
}

```

Applies the given [transform] function to each element of the original collection and appends the results to the given [destination].

```

public inline fun <T, R, C : MutableCollection<in R>>
Iterable<T>.mapTo(destination:
C, transform: (T) -> R): C {
 for (item in this)
 destination.add(transform(item))
 return
destination
}

```

Returns a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

```

public fun <T>
Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
 return IndexingIterable { iterator() }
}

```

Returns a list containing only distinct elements from the given collection.

```

Among equal elements of the
given collection, only the first one will be present in the resulting list.
The elements in the resulting list are in the
same order as they were in the source collection.
@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun <T>
Iterable<T>.distinct():
List<T> {
 return this.toMutableSet().toList()
}

```

Returns a list containing only elements from the given collection having distinct keys returned by the given [selector] function.

```

Among elements of the given collection with equal
keys, only the first one will be present in the resulting list.
The elements in the resulting list are in the same order
as they were in the source collection.
@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <T, K>
Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
 val set = HashSet<K>()
 val list = ArrayList<T>()

```

```

for (e in this) {
 val key = selector(e)
 if (set.add(key))
 list.add(e)
}
return list
}

* Returns a set containing all elements that are contained by both this collection and the specified collection.
The returned set preserves the element iteration order of the original collection.
To get a set containing all elements that are contained at least in one of these collections use [union].

public infix fun
<T> Iterable<T>.intersect(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.retainAll(other)
 return set
}

* Returns a set containing all elements that are contained by this collection and not contained
by the specified collection.
The returned set preserves the element iteration order of the original
collection.

public infix fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T> {
 val set =
this.toMutableSet()
 set.removeAll(other)
 return set
}

* Returns a new [MutableSet] containing all
distinct elements from the given collection.
The returned set preserves the element iteration order of the
original collection.

public fun <T> Iterable<T>.toMutableSet(): MutableSet<T> {
 return when (this) {
 is Collection<T> -> LinkedHashSet(this)
 else -> toCollection(LinkedHashSet<T>())
 }
}

* Returns a set containing all distinct elements from both collections.
The returned set preserves the element iteration order of the original collection.
Those elements of the [other]
collection that are unique are iterated in the end
in the order of the [other] collection.
To get a set
containing all elements that are contained in both collections use [intersect].

public infix fun <T>
Iterable<T>.union(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.addAll(other)
 return set
}

* Returns `true` if all elements match the given [predicate].
Note that if the collection
contains no elements, the function returns `true` because there are no elements in it that _do not_ match the
predicate.
See a more detailed explanation of this logic concept in ["Vacuous
truth"](https://en.wikipedia.org/wiki/Vacuous_truth) article.

@sample
samples.collections.Collections.Aggregates.all

public inline fun <T> Iterable<T>.all(predicate: (T) ->
Boolean): Boolean {
 if (this is Collection
 && isEmpty()) return true
 for (element in this) if (!predicate(element)) return false
 return true
}

* Returns `true` if collection has at least one element.

@sample
samples.collections.Collections.Aggregates.any

public fun <T> Iterable<T>.any(): Boolean {
 if (this is
Collection) return !isEmpty()
 return iterator().hasNext()
}

* Returns `true` if at least one element
matches the given [predicate].

@sample
samples.collections.Collections.Aggregates.anyWithPredicate

public inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean {
 if (this is Collection &&
isEmpty()) return false
 for (element in this) if (predicate(element)) return true
 return false
}

* Returns the number of elements in this collection.

public fun <T> Iterable<T>.count(): Int {
 if (this is
Collection) return size
 var count = 0
 for (element in this) checkCountOverflow(++count)
 return
count
}

* Returns the number of elements in this collection.

@kotlin.internal.InlineOnly
public inline fun <T>
Collection<T>.count(): Int {
 return size
}

* Returns the number of elements matching the given
[predicate].

public inline fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int {
 if (this is Collection
&& isEmpty()) return 0
 var count = 0
 for (element in this) if (predicate(element))
checkCountOverflow(++count)
 return count
}

* Accumulates value starting with [initial] value and
applying [operation] from left to right
to current accumulator value and each element.
Returns the
specified [initial] value if the collection is empty.

@param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.

public inline fun <T, R>
Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
 var accumulator = initial
 for (element
in this) accumulator = operation(accumulator, element)
 return accumulator
}

* Accumulates value
starting with [initial] value and applying [operation] from left to right
to current accumulator value and each
element with its index in the original collection.
Returns the specified [initial] value if the collection is
empty.

@param [operation] function that takes the index of an element, current accumulator value
and the element itself, and calculates the next accumulator value.

public inline fun <T, R>
Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
 var index = 0
 var

```

```

accumulator = initial\n for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator,
element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified
[initial] value if the list is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R>
List<T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n var accumulator = initial\n if (!isEmpty()) {\n
val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n accumulator =
operation(iterator.previous(), accumulator)\n }\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the
original list and current accumulator value.\n * \n * Returns the specified [initial] value if the list is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself\n * and current accumulator
value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> List<T>.foldRightIndexed(initial:
R, operation: (index: Int, T, acc: R) -> R): R {\n var accumulator = initial\n if (!isEmpty()) {\n val iterator =
listIterator(size)\n while (iterator.hasPrevious()) {\n val index = iterator.previousIndex()\n
accumulator = operation(index, iterator.previous(), accumulator)\n }\n }\n return accumulator\n}\n\n/**\n *
Performs the given [action] on each element.\n */\n@kotlin.internal.HidesMembers\npublic inline fun <T>
Iterable<T>.forEach(action: (T) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the
given [action] on each element, providing sequential index with the element.\n * @param [action] function that
takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline
fun <T> Iterable<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n var index = 0\n for (item in this)
action(checkIndexOverflow(index++), item)\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws
NoSuchElementException if the collection is empty.\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun Iterable<Double>.max(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw
NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e =
iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun Iterable<Float>.max(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw
NoSuchElementException()\n var max = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n *
Returns the largest element.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun <T : Comparable<T>> Iterable<T>.max(): T {\n val iterator = iterator()\n if
(!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext())
{\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the
collection is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL
OADS")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.maxBy(selector:
(T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var maxVal = selector(maxElem)\n do
{\n val e = iterator.next()\n val v = selector(e)\n if (maxVal < v) {\n maxElem = e\n
maxVal = v\n }\n } while (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the first element
yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if

```



```
(iterator.hasNext()) return null\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var\n maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while\n (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by\n [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is\n empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->\nDouble): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var\n maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->\nFloat): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var\n maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all\n values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws\n NoSuchElementException if the\n collection is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.maxOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw\n NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v\n = selector(iterator.next())\n if (maxValue < v) {\n maxValue = v\n }\n }\n return\n maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to\n each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\ninline fun <T> Iterable<T>.maxOfOrNull(selector: (T) -> Double): Double? {\n val iterator = iterator()\n if\n (iterator.hasNext()) return null\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n collection or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the\n returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)\n-> Float): Float? {\n val iterator = iterator()\n if (iterator.hasNext()) return null\n var maxValue =\n selector(iterator.next())\n while (iterator.hasNext())\n {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return\n maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to\n each element in the collection or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
```

```

Iterable<T>.maxOrNull(selector: (T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return\n null\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v =\n selector(iterator.next())\n if (maxValue < v) {\n maxValue = v\n }\n }\n return\n maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the collection.\n *\n * @throws NoSuchElementException if the collection is empty.\n *\n */\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\n\nIterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n val iterator = iterator()\n if\n (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while\n (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided\n * [comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null`\n * if there are no elements.\n *\n */\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\n\ninline fun <T, R> Iterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n val\n iterator = iterator()\n if (!iterator.hasNext()) return null\n var maxValue = selector(iterator.next())\n while\n (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no\n * elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n */\n\n@SinceKotlin("1.4")\npublic fun\n\nIterable<Double>.maxOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var\n max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of\n * elements is\n * `NaN` returns `NaN`.\n *\n */\n\n@SinceKotlin("1.4")\npublic fun Iterable<Float>.maxOrNull(): Float? {\n val\n iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext())\n {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest\n * element or `null` if there are no elements.\n *\n */\n\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>>\n\nIterable<T>.maxOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max =\n iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided\n * [comparator].\n * \n * @throws NoSuchElementException if the collection is empty.\n *\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER\nLOADS")\npublic\n\nfun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T {\n val iterator = iterator()\n if\n (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext())\n {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no\n * elements.\n *\n */\n\n@SinceKotlin("1.4")\npublic fun <T> Iterable<T>.maxWithOrNull(comparator: Comparator<in\nT>): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while\n (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n *\n */\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\nDS")\npublic\n\nfun Iterable<Double>.min(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw\n NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e =

```

```

iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun Iterable<Float>.min(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw\n NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e =\n iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException\n * if the collection is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun <T : Comparable<T>> Iterable<T>.min(): T {\n val iterator = iterator()\n if\n (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext())\n {\n val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element\n * yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the collection is\n * empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minBy(selector: (T) -> R): T {\n val iterator =\n iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minElem = iterator.next()\n if\n (!iterator.hasNext())\n return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v =\n selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while\n (iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the\n * given function or `null` if there are no elements.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if\n (!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var\n minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue >\n v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return\n minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result\n * is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->\n Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var\n minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all\n * values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values\n * produced\n * by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the\n * collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->\n Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var\n minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all\n * values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws\n * NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun <T, R : Comparable<R>> Iterable<T>.minOf(selector: (T) -> R): R {
 val iterator = iterator()
 if (!iterator.hasNext()) throw NoSuchElementException()
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 if (minValue > v) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Double): Double? {
 val iterator = iterator()
 if (!iterator.hasNext()) return null
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 minValue = minOf(minValue, v)
 }
 return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Float): Float? {
 val iterator = iterator()
 if (!iterator.hasNext()) return null
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 minValue = minOf(minValue, v)
 }
 return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Iterable<T>.minOfOrNull(selector: (T) -> R): R? {
 val iterator = iterator()
 if (!iterator.hasNext()) return null
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 if (minValue > v) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the collection.
@throws NoSuchElementException if the collection is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {
 val iterator = iterator()
 if (!iterator.hasNext()) throw NoSuchElementException()
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
 val iterator = iterator()
 if (!iterator.hasNext()) return null
 var minValue = selector(iterator.next())
 while (iterator.hasNext()) {
 val v = selector(iterator.next())
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest element or `null` if there are no elements.
If any of elements is `NaN` returns `NaN`.

@SinceKotlin("1.4")
public fun Iterable<Double>.minOrNull(): Double? {
 val iterator = iterator()
 if (!iterator.hasNext()) return null
 var min = iterator.next()
 while (iterator.hasNext()) {
 val e = iterator.next()
 min = minOf(min, e)
 }
 return min
}

Returns the smallest element or `null` if there are no elements.
If any of elements is `NaN` returns `NaN`.

@SinceKotlin("1.4")
public fun Iterable<Float>.minOrNull(): Float? {
 val iterator = iterator()
 if

```

```

(!iterator.hasNext()) return
null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min =
minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n */\npublic fun <T : Comparable<T>> Iterable<T>.minOrNull(): T? {\n val iterator =
iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n
val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having
the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the
collection is empty.\n
*\n * @SinceKotlin("1.7")\n */\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T {\n val iterator = iterator()\n
if (!iterator.hasNext()) throw NoSuchElementException()\n var
min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(min,
e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to
the provided [comparator] or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n */\npublic fun <T>
Iterable<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e =
iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if
the collection has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
*\n */\npublic fun
<T> Iterable<T>.none(): Boolean {\n if (this is Collection) return isEmpty()\n return
!iterator().hasNext()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n
*\n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n */\npublic inline fun <T>
Iterable<T>.none(predicate: (T) -> Boolean): Boolean {\n if (this is Collection && isEmpty()) return true\n for
(element in this) if (predicate(element)) return false\n return true\n}\n\n/**\n * Performs the given [action] on each
element and returns the collection itself afterwards.\n
*\n * @SinceKotlin("1.1")\n */\npublic inline fun <T, C :
Iterable<T>> C.onEach(action: (T) -> Unit): C {\n return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the collection itself afterwards.\n * \n * @param [action] function that takes the index of an element and the
element itself\n * and performs the action on the element.\n
*\n * @SinceKotlin("1.4")\n */\npublic inline fun <T, C :
Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {\n return apply { forEachIndexed(action)
}\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to
current accumulator value and each element.\n * \n * \n * Throws an exception if this collection is empty. If the
collection can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver
is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and
calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n */\npublic inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {\n val iterator = this.iterator()\n
if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")\n var
accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator = operation(accumulator,
iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element
with its index in the original collection.\n * \n * \n * Throws an exception if this collection is empty. If the collection can
be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is
empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the
element itself,\n * and calculates the next accumulator value.\n
*\n * @sample
samples.collections.Collections.Aggregates.reduce\n
*\n */\npublic inline fun <S, T : S>
Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n val iterator = this.iterator()\n if
(!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")\n var index =
1\n var accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator =
operation(checkIndexOverflow(index++),

```

```

 accumulator, iterator.next())\n } \n return accumulator\n}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element with its index
in the original collection.\n * \n * Returns `null` if the collection is empty.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <S, T : S> Iterable<T>.reduceIndexedOrNull(operation: (index: Int,
acc: S, T) -> S): S? {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return null\n var index = 1\n var
accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator =
operation(checkIndexOverflow(index++), accumulator, iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Returns `null` if the collection is empty.\n * \n * @param [operation]
function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n val iterator = this.iterator()\n if
(!iterator.hasNext()) return null\n var accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n
accumulator = operation(accumulator, iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n
* Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use
[reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*\npublic inline fun <S, T : S>
List<T>.reduceRight(operation: (T, acc: S) -> S): S {\n val iterator = listIterator(size)\n if
(!iterator.hasPrevious())\n throw UnsupportedOperationException("Empty list can't be reduced.")\n var
accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n accumulator =
operation(iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the last element and applying [operation] from right to left\n * to each element with its index in the original list
and current accumulator
value.\n * \n * Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n
*\npublic
inline fun <S, T : S> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n val iterator =
listIterator(size)\n if (!iterator.hasPrevious())\n throw UnsupportedOperationException("Empty list can't be
reduced.")\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n val index =
iterator.previousIndex()\n accumulator = operation(index, iterator.previous(), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element
and applying [operation] from right to left\n * to each element with its index in the original list and current
accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * @param [operation] function that takes the
index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int,
T, acc: S) -> S): S? {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n return null\n var
accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n val index = iterator.previousIndex()\n
accumulator = operation(index, iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n

```

\* to each element and current accumulator value.\n \* \n \* Returns `null` if the list is empty.\n \* \n \* @param [operation] function that takes an element and current accumulator value,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n return null\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n accumulator = operation(iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```

```

*\n@SinceKotlin("1.4")\npublic inline fun <T, R> Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n val estimatedSize = collectionSizeOrDefault(9)\n if (estimatedSize == 0) return listOf(initial)\n val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```

```

*\n@SinceKotlin("1.4")\npublic inline fun <T, R> Iterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n val estimatedSize = collectionSizeOrDefault(9)\n if (estimatedSize == 0) return listOf(initial)\n val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n var index = 0\n var accumulator = initial\n for (element in this) {\n accumulator = operation(index++, accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this collection.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return emptyList()\n var accumulator: S = iterator.next()\n val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\n while (iterator.hasNext()) {\n accumulator = operation(accumulator, iterator.next())\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator value that starts with the first element of this collection.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

```

```

*\n@SinceKotlin("1.4")\npublic inline fun <S, T : S> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return emptyList()\n

```

```

 var accumulator: S = iterator.next()\n val result = ArrayList<S>(collectionSizeOrDefault(10)).apply {
 add(accumulator) }\n var index = 1\n while (iterator.hasNext()) {\n accumulator = operation(index++,
 accumulator, iterator.next())\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
 containing successive accumulation values generated by applying [operation] from left to right\n * to each element
 and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
 function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
 [operation] function that takes current accumulator value and an element, and calculates the next accumulator
 value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
 *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
 Iterable<T>.scan(initial: R, operation:
 (acc: R, T) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing
 successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in
 the original collection and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value
 passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting
 list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
 element itself, and calculates the next accumulator value.\n * \n * @sample
 samples.collections.Collections.Aggregates.scan\n
 *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
 Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n return
 runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values
 produced by [selector] function applied to each element in the collection.\n *\n@Deprecated("Use sumOf
 instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline
 fun <T> Iterable<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
 selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
 applied to each element in the collection.\n *\n@Deprecated("Use sumOf instead.",
 ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
 Iterable<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this)
 {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
 [selector] function applied to each element in the collection.\n
 *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
 ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic
 inline fun <T> Iterable<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for
 (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
 produced by [selector] function applied to each element in the collection.\n
 *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
 ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
 Iterable<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum +=
 selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
 applied to each element in the collection.\n
 *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
 ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic
 inline fun <T> Iterable<T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element
 in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced
 by [selector] function applied to each element in the collection.\n
 *\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
 ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
 s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {\n
 var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return

```



`sum`  
`sum` Returns the sum of all values produced by [selector] function applied to each element in the collection.  
`sum`  
`sum` Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.  
`sum`  
`sum` Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.  
`sum`  
`sum` Splits this collection into a list of lists each not exceeding the given [size].  
`sum`  
`sum` Splits this collection into several lists each not exceeding the given [size] and applies the given [transform] function to an each.  
`sum`  
`sum` Returns a list containing all elements of the original collection without the first occurrence of the given [element].  
`sum`  
`sum` Returns a list containing all elements of the original collection except the elements contained in the given [elements] array.  
`sum`  
`sum` Returns a list containing all elements of the original collection except the elements contained in the given [elements] collection.  
`sum`  
`sum` Returns a list containing all elements of the original collection except the elements contained in the given [elements] sequence.  
`sum`  
`sum` Returns a list containing all elements of the original collection without the first occurrence of the given [element].  
`sum`  
`sum` Splits the original collection into pair of lists, where \*first\* list contains elements for which [predicate] yielded `true`, while \*second\* list contains elements for which [predicate] yielded `false`.  
`sum`

```

Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val second = ArrayList<T>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\npublic operator fun <T> Iterable<T>.plus(element: T): List<T> {\n if (this is Collection) return this.plus(element)\n val result = ArrayList<T>()\n result.addAll(this)\n result.add(element)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\npublic operator fun <T> Collection<T>.plus(element: T): List<T> {\n val result = ArrayList<T>(size + 1)\n result.addAll(this)\n result.add(element)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] array.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Array<out T>): List<T> {\n if (this is Collection) return this.plus(elements)\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.\n */\npublic operator fun <T> Collection<T>.plus(elements: Collection<T>): List<T> {\n val result = ArrayList<T>(this.size + elements.size)\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Iterable<T>): List<T> {\n if (this is Collection) return this.plus(elements)\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.\n */\npublic operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {\n if (elements is Collection) {\n val result = ArrayList<T>(this.size + elements.size)\n result.addAll(this)\n result.addAll(elements)\n return result\n } else {\n val result = ArrayList<T>(this)\n result.addAll(elements)\n return result\n }\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] sequence.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] sequence.\n */\npublic operator fun <T> Collection<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>(this.size + 10)\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/n/**\n * Returns a list of snapshots of the window of the given [size]\n * sliding along this collection with the given [step], where each\n * snapshot is a list.\n * Several last lists may have fewer elements than the given [size].\n * Both [size] and [step] must be positive and can be greater than the number of elements in this collection.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved\n * @sample samples.collections.Sequences.Transformations.takeWindows\n */\n@SinceKotlin("1.2")\npublic fun <T> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<List<T>> {\n checkWindowSizeStep(size, step)\n if (this is RandomAccess && this is List) {\n val thisSize = this.size\n val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result = ArrayList<List<T>>(resultCapacity)\n var index = 0\n while (index in 0 until thisSize) {\n val windowSize = size.coerceAtMost(thisSize - index)\n if (windowSize < size && !partialWindows) break\n result.add(List(windowSize) { this[it

```

```

+ index] })\n index += step\n }\n return result\n }\n val result = ArrayList<List<T>>()\n windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {\n result.add(it)\n }\n return result\n }\n /**\n * Returns a list of results of applying the given [transform] function to\n * an each list representing a view over the window of the given [size]\n * sliding along this collection with the given [step].\n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * Both [size] and [step] must be positive and can be greater than the number of elements in this collection.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward\n * by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample\n samples.collections.Sequences.Transformations.averageWindows\n *\n @SinceKotlin("1.2")\n \n public fun <T, R> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): List<R> {\n checkWindowSizeStep(size, step)\n if (this is RandomAccess && this is List) {\n val thisSize = this.size\n val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result = ArrayList<R>(resultCapacity)\n val window = MovingSubList(this)\n var index = 0\n while (index in 0 until thisSize) {\n val windowSize = size.coerceAtMost(thisSize - index)\n if (!partialWindows && windowSize < size) break\n window.move(index, index + windowSize)\n result.add(transform(window))\n index += step\n }\n return result\n }\n val result = ArrayList<R>()\n windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {\n result.add(transform(it))\n }\n return result\n }\n /**\n * Returns a list of pairs built from the elements of `this` collection and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample\n samples.collections.Iterables.Operations.zipIterable\n *\n @public infix fun <T, R> Iterable<T>.zip(other: Array<out R>): List<Pair<T, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n }\n /**\n * Returns a list of values built from the elements of `this` collection and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample\n samples.collections.Iterables.Operations.zipIterableWithTransform\n *\n @public inline fun <T, R, V> Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n val arraySize = other.size\n val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in this) {\n if (i >= arraySize) break\n list.add(transform(element, other[i++]))\n }\n return list\n }\n /**\n * Returns a list of pairs built from the elements of `this` collection and [other] collection with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample\n samples.collections.Iterables.Operations.zipIterable\n *\n @public infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n }\n /**\n * Returns a list of values built from the elements of `this` collection and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * \n * The returned list has length of the shortest collection.\n * \n * @sample\n samples.collections.Iterables.Operations.zipIterableWithTransform\n *\n @public inline fun <T, R, V> Iterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n val first = iterator()\n val second = other.iterator()\n val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10)))\n while (first.hasNext() && second.hasNext()) {\n list.add(transform(first.next(), second.next()))\n }\n return list\n }\n /**\n * Returns a list of pairs of each two adjacent elements in this collection.\n * \n * The returned list is empty if this collection contains less than two elements.\n * \n * @sample\n samples.collections.Collections.Transformations.zipWithNext\n *\n @SinceKotlin("1.2")\n \n @public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {\n return zipWithNext { a, b -> a to b }\n }\n /**\n * Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent elements in this collection.\n * \n * The returned list is empty if this collection contains less than two elements.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n *\n@SinceKotlin("1.2")\npublic
inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> R): List<R> {\n val iterator = iterator()\n if
(iterator.hasNext()) return emptyList()\n val result = mutableListOf<R>()\n var current = iterator.next()\n
while (iterator.hasNext()) {\n val next = iterator.next()\n result.add(transform(current, next))\n current
= next\n }\n return result\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and
using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-
negative value of [limit], in which case only the first [limit]\n * elements will be appended,
followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n *\npublic fun <T, A : Appendable>
Iterable<T>.joinTo(buffer: A, separator: CharSequence = '\n', prefix: CharSequence = "\n", postfix: CharSequence
= "\n", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {\n
buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n
 if (limit < 0 || count <= limit) {\n buffer.appendElement(element, transform)\n } else break\n }\n if
(limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n *
Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if
supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which
case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to
"...").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n *\npublic fun <T>
Iterable<T>.joinToString(separator: CharSequence = '\n', prefix: CharSequence = "\n", postfix: CharSequence =
"\n", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {\n
return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Returns this
collection as an [Iterable].\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.asIterable():
Iterable<T> {\n return this\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original collection returning
its elements when being iterated.\n * \n * @sample
samples.collections.Sequences.Building.sequenceFromCollection\n *\npublic fun <T> Iterable<T>.asSequence():
Sequence<T> {\n return Sequence
{ this.iterator() }\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun Iterable<Byte>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n
checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the collection.\n *\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun
Iterable<Short>.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Iterable<Int>.average(): Double {\n var sum: Double =
0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun Iterable<Long>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n
checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the collection.\n *\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun
Iterable<Float>.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n
sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the collection.\n
*\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Iterable<Double>.average():
Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n
checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
an average value of elements in the collection.\n

```

```

the sum of all elements in the collection.\n *^@kotlin.jvm.JvmName("sumOfByte")\npublic fun
Iterable<Byte>.sum(): Int {\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n return
sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*^@kotlin.jvm.JvmName("sumOfShort")\npublic fun Iterable<Short>.sum(): Int {\n var sum: Int = 0\n for
(element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the
collection.\n *^@kotlin.jvm.JvmName("sumOfInt")\npublic fun Iterable<Int>.sum(): Int {\n var sum: Int = 0\n
for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements
in the collection.\n *^@kotlin.jvm.JvmName("sumOfLong")\npublic fun Iterable<Long>.sum(): Long {\n var
sum: Long = 0L\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the
sum of all elements in the collection.\n *^@kotlin.jvm.JvmName("sumOfFloat")\npublic fun
Iterable<Float>.sum(): Float {\n var sum: Float = 0.0f\n for (element in this) {\n sum += element\n }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*^@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Iterable<Double>.sum(): Double {\n var sum: Double
= 0.0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n"/\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^@kotlin.package kotlin.collections\n\nimport
kotlin.comparisons.naturalOrder\n\nimport
kotlin.random.Random\n\nimport kotlin.js.arrayBufferIsView\n\n/**\n * Returns the array if it's not `null`, or an
empty array otherwise.\n * @sample samples.collections.Arrays.Usage.arrayOrEmpty\n
*^@kotlin.internal.InlineOnly\n\npublic actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = this ?:
emptyArray<T>()\n\n/**\n * Returns a *typed* array containing all of the elements of this collection.\n *^@kotlin
* Allocates an array of runtime type `T` having its size equal to the size of this collection\n * and populates the array
with the elements of this collection.\n * @sample
samples.collections.Collections.Collections.collectionToTypedArray\n *^@kotlin.internal.InlineOnly\n\npublic
actual inline fun <T> Collection<T>.toArray(): Array<T> =
copyToArray(this)\n\n@JsName("copyToArray")\n@PublishedApi\n\ninternal fun <T> copyToArray(collection:
Collection<T>): Array<T> {\n return if (collection.asDynamic().toArray !== undefined)\n collection.asDynamic().toArray().unsafeCast<Array<T>>()\n
 else\n copyToArrayImpl(collection).unsafeCast<Array<T>>()\n}\n\n@JsName("copyToArrayImpl")\n\ninternal actual fun
copyToArrayImpl(collection: Collection<*>): Array<Any?> {\n val array = emptyArray<Any?>()\n val iterator
= collection.iterator()\n while (iterator.hasNext())\n array.asDynamic().push(iterator.next())\n return
array\n}\n\n@JsName("copyToExistingArrayImpl")\n\ninternal actual fun <T> copyToArrayImpl(collection:
Collection<*>, array: Array<T>): Array<T> {\n if (array.size < collection.size)\n return
copyToArrayImpl(collection).unsafeCast<Array<T>>()\n val iterator = collection.iterator()\n var index = 0\n
while (iterator.hasNext()) {\n array[index++] = iterator.next().unsafeCast<T>()\n }\n if (index < array.size)
{\n array[index] = null.unsafeCast<T>()\n }\n return array\n}\n\n\n/**\n * Returns an immutable list
containing only the specified object [element].\n *^@kotlin\n\npublic
fun <T> listOf(element: T): List<T> =
arrayListOf(element)\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n\ninternal actual
inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {\n return
ArrayList<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.Inlin
eOnly\n\ninternal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit):
List<E> {\n checkBuilderCapacity(capacity)\n return
ArrayList<E>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Returns an immutable set containing only the
specified object [element].\n *^@kotlin\n\npublic fun <T> setOf(element: T): Set<T> =
hashSetOf(element)\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n\ninternal actual inline
fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {\n return

```

```

LinkedHashSet<E>().apply(builderAction).build()\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
 actual inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>().->Unit): Set<E> {\n return
LinkedHashSet<E>(capacity).apply(builderAction).build()\n\n\n/**\n * Returns an immutable map, mapping
only the specified key to the\n * specified value.\n */\npublic fun <K, V> mapOf(pair: Pair<K, V>): Map<K, V> =
HashMapOf(pair)\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline
fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>().->Unit): Map<K, V> {\n return
LinkedHashMap<K,
V>().apply(builderAction).build()\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
actual inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>().->Unit):
Map<K, V> {\n return LinkedHashMap<K, V>(capacity).apply(builderAction).build()\n\n\n\n/**\n * Fills the
list with the provided [value].\n * Each element in the list gets
replaced with the [value].\n */\n@SinceKotlin("1.2")\npublic actual fun <T> MutableList<T>.fill(value: T): Unit
{\n for (index in 0..lastIndex) {\n this[index] = value\n }\n\n\n/**\n * Randomly shuffles elements in this
list.\n * See: https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin("1.2")\npublic actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)\n\n\n/**\n * Returns a new list with the elements of this list randomly shuffled.\n */\n@SinceKotlin("1.2")\npublic actual fun
<T> Iterable<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }\n\n\n/**\n * Sorts elements in the list in-
place according to their natural sort order.\n * The sort is _stable_. It means that equal elements preserve their
order relative to each other after sorting.\n * @sample samples.collections.Collections.Sorting.sortMutableList\n */\npublic actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {\n CollectionsSort(this,
naturalOrder())\n}\n\n\n/**\n * Sorts elements in the list in-place according to the order specified with
[comparator].\n * The sort is _stable_. It means that equal elements preserve their order relative to each other
after sorting.\n * @sample samples.collections.Collections.Sorting.sortMutableListWith\n */\npublic actual fun
<T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit {\n CollectionsSort(this,
comparator)\n}\n\n\nprivate fun <T> CollectionsSort(list: MutableList<T>, comparator: Comparator<in T>) {\n if
(list.size <= 1) return\n\n val array = copyToArray(list)\n sortArrayWith(array, comparator)\n\n for (i in 0 until
array.size) {\n list[i] = array[i]\n }\n\n\ninternal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int):
Array<T> {\n return
arrayOfNulls<Any>(size).unsafeCast<Array<T>>()\n}\n\n\n@SinceKotlin("1.3")\n@PublishedApi\n@JsName("arrayCopy")\ninternal fun <T> arrayCopy(source: Array<out T>, destination:
Array<in T>, destinationOffset: Int, startIndex: Int, endIndex: Int) {\n
AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)\n val rangeSize = endIndex - startIndex\n
AbstractList.checkRangeIndexes(destinationOffset, destinationOffset + rangeSize, destination.size)\n\n if
(arrayBufferIsView(destination) && arrayBufferIsView(source)) {\n val subrange =
source.asDynamic().subarray(startIndex, endIndex)\n destination.asDynamic().set(subrange,
destinationOffset)\n } else {\n if (source !== destination || destinationOffset <= startIndex) {\n for
(index in 0 until rangeSize) {\n destination[destinationOffset + index] = source[startIndex + index]\n
 }\n } else {\n for (index in rangeSize - 1 downTo 0) {\n destination[destinationOffset + index] =
source[startIndex + index]\n }\n }\n }\n\n\n// no singleton map implementation in js, return map as
is\n@Suppress("NOTHING_TO_INLINE")\ninternal
 actual inline fun <K, V> Map<K, V>.toSingletonMapOrSelf(): Map<K, V> =
this\n\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline fun <K, V> Map<out K,
V>.toSingletonMap(): Map<K, V> = this.toMutableMap()\n\n\n@Suppress("NOTHING_TO_INLINE")\ninternal
actual inline fun <T> Array<out T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =\n if
(isVarargs)\n // no need to copy vararg array in JS\n this\n else\n this.copyOfOf()\n\n\n\n@PublishedApi\ninternal actual fun checkIndexOverflow(index: Int): Int {\n if (index < 0)
{\n throwIndexOverflow()\n }\n return index\n}\n\n\n@PublishedApi\ninternal actual fun

```

```

checkCountOverflow(count: Int): Int {
 if (count < 0) {
 throwCountOverflow()
 }
 return count
}

```

JS map and set implementations do not make use of capacities or load factors.

```

@PublishedApi
internal fun mapCapacity(expectedSize: Int) = expectedSize

```

Checks a collection builder function capacity argument. In JS no validation is made in Map/Set constructor yet.

```

@SinceKotlin("1.3")
@PublishedApi
internal fun checkBuilderCapacity(capacity: Int) {
 require(capacity >= 0) { "capacity must be non-negative." }
}

```

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")
package kotlin.collections

```

Returns the given iterator itself. This allows to use an instance of iterator in a for loop.

```

@sample samples.collections.Iterators.iterator

```

```

@kotlin.internal.InlineOnly
public inline operator fun <T> Iterator<T>.iterator(): Iterator<T> = this

```

Returns an [Iterator] that wraps each element produced by the original iterator into an [IndexedValue] containing the index of that element and the element itself.

```

@sample samples.collections.Iterators.withIndexIterator

```

```

public fun <T> Iterator<T>.withIndex(): Iterator<IndexedValue<T>> = IndexingIterator(this)

```

Performs the given [operation] on each element of this [Iterator].

```

@sample samples.collections.Iterators.forEachIterator

```

```

public inline fun <T> Iterator<T>.forEach(operation: (T) -> Unit): Unit {
 for (element in this) operation(element)
}

```

Returns an [Iterator] that wraps each element produced by the original iterator into an [IndexedValue], counting index from zero.

```

internal class IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {
 private var index = 0
 final override fun hasNext(): Boolean = iterator.hasNext()
 final override fun next(): IndexedValue<T> = IndexedValue(checkIndexOverflow(index++), iterator.next())
}

```

Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ComparisonsKt")
package kotlin.comparisons

```

NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt. See: <https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib>

```

import kotlin.random.*

```

Returns the greater of two values. If values are equal, returns the first one.

```

@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T): T

```

Returns the greater of two values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Byte, b: Byte): Byte

```

Returns the greater of two values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Short, b: Short): Short

```

Returns the greater of two values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int): Int

```

Returns the greater of two values. If either value is NaN, returns NaN.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long): Long

```

Returns the greater of two values. If either value is NaN, returns NaN.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Float, b: Float): Float

```

Returns the greater of two values. If either value is NaN, returns NaN.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Double, b: Double): Double

```

Returns the greater of three values. If there are multiple equal maximal values, returns the first of them.

```

@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T

```

Returns the greater of three values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte

```

Returns the greater of three values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Short, b: Short, c: Short): Short

```

Returns the greater of three values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int, c: Int): Int

```

Returns the greater of three values.

```

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long, c: Long): Long

```

Returns the greater of three values. If

any value is `NaN`, returns `NaN`.

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline
fun maxOf(a: Float, b: Float, c: Float): Float {
 * Returns the greater of three values.
 * If any value is `NaN`, returns `NaN`.
}

```

Double, b: Double, c: Double): Double

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun maxOf(a: Double, b: Double, c: Double): Double {
 * Returns the greater of three values according to the order specified by the given [comparator].
 * If there are multiple equal maximal values, returns the first of them.
}

```

```

@SinceKotlin("1.1")@public fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {
 return maxOf(a, maxOf(b, c, comparator), comparator)
}

```

```

@SinceKotlin("1.1")@public fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T {
 return if (comparator.compare(a, b) >= 0) a else b
}

```

```

@SinceKotlin("1.4")@public expect fun <T : Comparable<T>> maxOf(a: T, vararg other: T): T {
 * Returns the greater of the given values.
 * If there are multiple equal maximal values, returns the first of them.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Byte, vararg other: Byte): Byte {
 * Returns the greater of the given values.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Short, vararg other: Short): Short {
 * Returns the greater of the given values.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Int, vararg other: Int): Int {
 * Returns the greater of the given values.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Long, vararg other: Long): Long {
 * Returns the greater of the given values.
 * If any value is `NaN`, returns `NaN`.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Float, vararg other: Float): Float {
 * Returns the greater of the given values.
 * If any value is `NaN`, returns `NaN`.
}

```

```

@SinceKotlin("1.4")@public expect fun maxOf(a: Double, vararg other: Double): Double {
 * Returns the greater of the given values according to the order specified by the given [comparator].
 * If there are multiple equal maximal values, returns the first of them.
}

```

```

@SinceKotlin("1.4")@public fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T {
 var max = a
 for (e in other) if (comparator.compare(max, e) < 0) max = e
 return max
}

```

```

@SinceKotlin("1.1")@public expect fun <T : Comparable<T>> minOf(a: T, b: T): T {
 * Returns the smaller of two values.
 * If values are equal, returns the first one.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Byte, b: Byte): Byte {
 * Returns the smaller of two values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Short, b: Short): Short {
 * Returns the smaller of two values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Int, b: Int): Int {
 * Returns the smaller of two values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Long, b: Long): Long {
 * Returns the smaller of two values.
 * If either value is `NaN`, returns `NaN`.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Float, b: Float): Float {
 * Returns the smaller of two values.
 * If either value is `NaN`, returns `NaN`.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Double, b: Double): Double {
 * Returns the smaller of three values.
 * If there are multiple equal minimal values, returns the first of them.
}

```

```

@SinceKotlin("1.1")@public expect fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {
 * Returns the smaller of three values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {
 * Returns the smaller of three values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Short, b: Short, c: Short): Short {
 * Returns the smaller of three values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Int, b: Int, c: Int): Int {
 * Returns the smaller of three values.
}

```

```

@SinceKotlin("1.1")@kotlin.internal.InlineOnly@public expect inline fun minOf(a: Long, b: Long, c: Long): Long {
 * Returns the smaller of three values.
 * If
}

```



```

any value is `NaN`, returns `NaN`.
inline fun minOf(a: Float, b: Float, c: Float): Float {
 * Returns the smaller of three values.
 * If any value is `NaN`, returns `NaN`.
}

inline fun minOf(a: Double, b: Double, c: Double): Double {
 * Returns the smaller of three values according to the order
 specified by the given [comparator].
 * If there are multiple equal
 minimal values, returns the first of them.
}

inline fun <T> minOf(a: T, b: T, c: T,
 comparator: Comparator<in T>): T {
 return minOf(a, minOf(b, c, comparator), comparator)
}

inline fun minOf(a: T, b: T, comparator:
 Comparator<in T>): T {
 return if (comparator.compare(a, b) <= 0) a else b
}

inline fun minOf(a: T, vararg other: T): T {
 * Returns the smaller of
 the given values.
 * If there are multiple equal minimal values, returns the first of them.
}

inline fun minOf(a: Byte, vararg
 other: Byte): Byte {
 * Returns the smaller of the given values.
}

inline fun minOf(a:
 Short, vararg other: Short): Short {
 * Returns the smaller of the given values.
}

inline fun minOf(a: Int, vararg other: Int): Int {
 * Returns the smaller of
 the given values.
}

inline fun minOf(a: Long, vararg other: Long):
 Long {
 * Returns the smaller of the given values.
 * If any value is `NaN`, returns `NaN`.
}

inline fun minOf(a: Float, vararg other: Float):
 Float {
 * Returns the
 smaller of the given values.
 * If any value is `NaN`, returns `NaN`.
}

inline fun minOf(a: Double, vararg other: Double):
 Double {
 * Returns the smaller of the given values
 according to the order specified by the given [comparator].
 * If there are multiple equal minimal values,
 returns the first of them.
}

inline fun <T> minOf(a: T, vararg other: T, comparator:
 Comparator<in T>): T {
 var
 min = a
 for (e in other) if (comparator.compare(min, e) > 0) min = e
 return min
}

/* Copyright
2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

package
kotlin.collections
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

Returns the first non-null value produced by
[transform] function being applied to entries of this map in iteration order,
or throws
[NoSuchElementException] if no non-null value was produced.
@sample
samples.collections.Collections.Transformations.firstNotNullOf

inline fun <K, V, R : Any> Map<out K, V>.firstNotNullOf(transform: (Map.Entry<K, V>) -> R?): R {
 return
 firstNotNullOfOrNull(transform) ?: throw NoSuchElementException("No element of the map was transformed to a
 non-null value.")
}

Returns the first non-null value produced by [transform] function being applied to
entries of this map in iteration order,
or `null` if no non-null value was produced.
@sample
samples.collections.Collections.Transformations.firstNotNullOf

inline fun <K, V, R : Any> Map<out K,
V>.firstNotNullOfOrNull(transform: (Map.Entry<K, V>) -> R?): R? {
 for (element in this) {
 val result =
 transform(element)
 if (result != null) {
 return result
 }
 }
 return null
}

Returns a [List] containing all key-value pairs.
public fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>>
{
 if (size == 0)
 return emptyList()
 val iterator = entries.iterator()
 if (!iterator.hasNext())
 return emptyList()
 val first = iterator.next()
 if (!iterator.hasNext())
 return listOf(first.toPair())
 val result =

```

```

ArrayList<Pair<K, V>>(size)\n result.add(first.toPair())\n do {\n result.add(iterator.next().toPair())\n }
while (iterator.hasNext())\n return result\n}\n\n/**\n * Returns a single list of all elements yielded from results of
[transform] function being invoked on each entry of original map.\n * \n * @sample
samples.collections.Maps.Transformations.flatMap\n */\npublic inline fun <K, V, R> Map<out K,
V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each entry of original map.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic
inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {\n
return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each entry of original map, to the given [destination].\n */\npublic inline fun
<K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>)
-> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n
}\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being
invoked on each entry of original map, to the given [destination].\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequenceTo")\npublic
inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform:
(Map.Entry<K, V>) -> Sequence<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each entry in the original map.\n * \n * @sample
samples.collections.Maps.Transformations.mapToList\n */\npublic inline fun <K, V, R> Map<out K,
V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {\n return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform]
function\n * to each entry in the original map.\n * \n * @sample
samples.collections.Maps.Transformations.mapNotNull\n */\npublic inline fun <K, V, R : Any> Map<out K,
V>.mapNotNull(transform:
(Map.Entry<K, V>) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies
the given [transform] function to each entry in the original map\n * and appends only the non-null results to the
given [destination].\n */\npublic inline fun <K, V, R : Any, C : MutableCollection<in R>> Map<out K,
V>.mapNotNullTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {\n forEach { element ->
transform(element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform]
function to each entry of the original map\n * and appends the results to the given [destination].\n */\npublic inline
fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.mapTo(destination: C, transform: (Map.Entry<K, V>)
-> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns
`true` if all entries match the given [predicate].\n * \n * Note that if the map contains no entries,
the function returns `true`\n * because there are no entries in it that do not match the predicate.\n * See a more
detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous_truth)
article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <K, V> Map<out K,
V>.all(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n if (isEmpty()) return true\n for (element in this)
if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if map has at least one entry.\n * \n *
@sample samples.collections.Collections.Aggregates.any\n */\npublic fun <K, V> Map<out K, V>.any(): Boolean
{\n return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one entry matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <K, V> Map<out K,
V>.any(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n if (isEmpty())
return false\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns the
number of entries in this map.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.count():

```



Returns the largest value according to the provided

[comparator] among all values produced by [selector] function applied to each entry in the map.

@throws NoSuchElementException if the map is empty.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
```

```
V>.maxOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n return
```

```
entries.maxOfWith(comparator, selector)\n}\n\n/**\n * Returns the largest value according to the provided
```

```
[comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
```

```
V>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n return
```

```
entries.maxOfWithOrNull(comparator,
```

```
selector)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided [comparator].\n
```

```
\n * @throws NoSuchElementException if the map is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <K, V> Map<out K, V>.maxWith(comparator:
```

```
Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {\n return entries.maxWith(comparator)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided [comparator] or `null` if there are no
```

```
entries.\n * \n * @since Kotlin("1.4")\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <K, V> Map<out K,
```

```
V>.maxWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n return
```

```
entries.maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given
```

```
function.\n * \n * @throws NoSuchElementException if the map is empty.\n
```

```
\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n
```

```
\n * \n * @since Kotlin("1.7")\n * \n * @kotlin.jvm.JvmName("minByOrThrow")\n * \n * @kotlin.internal.InlineOnly\n * \n * @Suppress("CONFLICTING_OVERLOADS")\n * \n * public inline fun <K, V, R : Comparable<R>> Map<out K,
```

```
V>.minBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V> {\n return entries.minBy(selector)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given function or `null` if there are no entries.\n
```

```
\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
```

```
\n * \n * @since Kotlin("1.4")\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <K, V, R : Comparable<R>> Map<out
```

```
K, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n return
```

```
entries.minByOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
```

```
function applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the
```

```
returned result is `NaN`.\n
```

```
\n * \n * @throws NoSuchElementException if the map is empty.\n
```

```
\n * \n * @throws NoSuchElementException if the map is empty.\n
```

```
\n * \n * @since Kotlin("1.4")\n * \n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n * @OverloadResolution\n * \n * ByLambdaReturnType\n * \n * @kotlin.internal.InlineOnly\n * \n * public inline fun <K, V> Map<out K, V>.minOf(selector:
```

```
(Map.Entry<K, V>) -> Double): Double {\n return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest
```

```
value among all values produced by [selector] function applied to each entry in the map.\n * \n * If any of values
```

```
produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
\n * \n * @throws NoSuchElementException
```

```
if the map is empty.\n
```

```
\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
```

```
(Map.Entry<K, V>) -> Float): Float {\n return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value
```

```
among all values produced by [selector]
```

```
function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
```

```

V>.minOf(selector: (Map.Entry<K, V>) -> R): R {
 return entries.minOf(selector)
}

```

Returns the smallest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```


```

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```


```

Since Kotlin 1.4

```


```

ExperimentalTypeInference

```


```

OverloadResolutionByLambdaReturnType

```


```

kotlin.internal.InlineOnly

```

public inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {
 return
entries.minOfOrNull(selector)
}

```

Returns the smallest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```


```

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```


```

Since Kotlin 1.4

```


```

ExperimentalTypeInference

```


```

OverloadResolutionByLambdaReturnType

```


```

kotlin.internal.InlineOnly

```

public inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {
 return
entries.minOfOrNull(selector)
}

```

Returns the smallest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```


```

Since Kotlin 1.4

```


```

ExperimentalTypeInference

```


```

OverloadResolutionByLambdaReturnType

```


```

kotlin.internal.InlineOnly

```

public inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {
 return entries.minOfOrNull(selector)
}

```

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map.

```


```

@throws NoSuchElementException if the map is empty.

```


```

Since Kotlin 1.4

```


```

ExperimentalTypeInference

```


```

OverloadResolutionByLambdaReturnType

```


```

kotlin.internal.InlineOnly

```

public inline fun <K, V, R> Map<out K,
V>.minOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {
 return
entries.minOfWith(comparator, selector)
}

```

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```


```

Since Kotlin 1.4

```


```

ExperimentalTypeInference

```


```

OverloadResolutionByLambdaReturnType

```


```

kotlin.internal.InlineOnly

```

public inline fun <K, V, R> Map<out K,
V>.minOfWithOrNull(comparator: Comparator<in
R>, selector: (Map.Entry<K, V>) -> R): R? {
 return entries.minOfWithOrNull(comparator,
selector)
}

```

Returns the first entry having the smallest value according to the provided [comparator].

```


```

@throws NoSuchElementException if the map is empty.

```


```

Since Kotlin 1.7

```


```

kotlin.jvm.JvmName("minWithOrThrow")

```


```

kotlin.internal.InlineOnly

```


```

Suppress("CONFLICTING\_OVERLOADS")

```


```

public inline fun <K, V> Map<out K, V>.minWith(comparator:
Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {
 return entries.minWith(comparator)
}

Returns the first entry having the smallest value according to the provided [comparator] or `null` if there are no entries.

```


```

Since Kotlin 1.4

```


```

kotlin.internal.InlineOnly

```


```

public inline fun <K, V> Map<out K,
V>.minWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {
 return
entries.minWithOrNull(comparator)
}

Returns `true` if the map has no entries.

```


```

@sample samples.collections.Collections.Aggregates.none

```


```

public fun <K, V> Map<out K, V>.none(): Boolean {
 return isEmpty()
}

Returns `true` if no entries match the given [predicate].

```


```

@sample samples.collections.Collections.Aggregates.noneWithPredicate

```


```

public inline fun <K, V> Map<out K,
V>.none(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {
 if (isEmpty()) return true
 for (element in this) if (predicate(element)) return false
 return true
}

Performs the given [action] on each entry and returns the map itself afterwards.

```


```

Since Kotlin 1.1

```


```

public inline fun <K, V, M : Map<out K, V>>
M.onEach(action: (Map.Entry<K, V>) -> Unit): M {
 return apply { for (element in this) action(element)
}
}

Performs the given [action] on each entry, providing sequential index with the entry, and returns the map itself afterwards.

```


```

@param [action] function that takes the index of an entry and the entry itself

performs

```
the action on the entry.\n *^@SinceKotlin("1.4")\npublic inline fun <K, V, M : Map<out K, V>>
M.onEachIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): M {\n return apply {
entries.forEachIndexed(action) }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original map returning
its entries when being iterated.\n *^@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.asIterable(): Iterable<Map.Entry<K, V>> {\n return entries\n}\n\n/**\n * Creates a [Sequence] instance that
wraps the original map returning its entries when being iterated.\n *^@public fun <K, V> Map<out K,
V>.asSequence(): Sequence<Map.Entry<K, V>> {\n return entries.asSequence()\n}\n\n"/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^@package kotlin.text\n\n// NOTE:
THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n//
See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n// 10 mappings totally\ninternal fun
Char.titlecaseImpl(): String {\n val uppercase = uppercase()\n if (uppercase.length > 1) {\n return if (this ==
"\u0149") uppercase else uppercase[0] + uppercase.substring(1).lowercase()\n }\n return
titlecaseChar().toString()\n}\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *^@package kotlin.text\n\n/**\n * Converts this character to lower case using Unicode
mapping rules of the invariant locale.\n *^@Deprecated("Use lowercaseChar() instead.")
ReplaceWith("lowercaseChar()")\n\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toLowerCase(): Char =
lowercaseChar()\n\n/**\n * Converts this character to lower
case using Unicode mapping rules of the invariant locale.\n *^*\n * This function performs one-to-one character
mapping.\n * To support one-to-many character mapping use the [lowercase] function.\n * If this character has no
mapping equivalent, the character itself is returned.\n *^*\n * @sample samples.text.Chars.lowercase\n\n*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
actual inline fun Char.lowercaseChar(): Char = lowercase()[0]\n\n/**\n * Converts this character to lower case
using Unicode mapping rules of the invariant locale.\n *^*\n * This function supports one-to-many character mapping,
thus the length of the returned string can be greater than one.\n * For example, ``\u0130'.lowercase()`` returns
``\u0069\u0307``,\n * where ``\u0130`` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character
(`\u0130`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n *^*\n * @sample samples.text.Chars.lowercase\n\n*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
actual inline fun Char.lowercase(): String = toString().asDynamic().toLowerCase().unsafeCast<String>()\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n *^*\n * @sample samples.text.Chars.uppercase\n\n*^@Deprecated("Use uppercaseChar() instead.")
ReplaceWith("uppercaseChar()")\n\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toUpperCase(): Char =
uppercaseChar()\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n *^*\n * This function performs one-to-one character mapping.\n * To support one-to-many character
mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself is
returned.\n *^*\n * @sample samples.text.Chars.uppercase\n\n*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
actual fun Char.uppercaseChar(): Char {\n val uppercase = uppercase()\n return if (uppercase.length > 1) this
else uppercase[0]\n}\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n *^*\n * This function supports one-to-many character mapping, thus the length of the returned string can be
greater than one.\n * For example, ``\uFB00'.uppercase()`` returns ``\u0046\u0046``,\n * where ``\uFB00`` is the
LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no upper case mapping, the result of
`toString()` of this char is returned.\n *^*\n * @sample samples.text.Chars.uppercase\n
```

```

*

@SinceKotlin("1.5")

@WasExperimental(ExperimentalStdlibApi::class)

@kotlin.internal.InlineOnly

public actual inline fun Char.toUpperCase(): String = toString().asDynamic().toUpperCase().unsafeCast<String>()

* Converts this character to title case using Unicode mapping rules of the invariant

locale.

* This function performs one-to-one character mapping.

* To support one-to-many character

mapping use the [titlecase] function.

* If this character has no mapping equivalent, the result of calling

[uppercaseChar] is returned.

* @sample samples.text.Chars.titlecase

@SinceKotlin("1.5")

public actual fun Char.titlecaseChar(): Char = titlecaseCharImpl()

* Returns `true` if this character is a Unicode

high-surrogate code unit (also known as leading-surrogate code unit).

public actual fun

Char.isHighSurrogate(): Boolean = this in

Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE

* Returns `true` if this character is a

Unicode low-surrogate code unit (also known as trailing-surrogate code unit).

public actual fun

Char.isLowSurrogate(): Boolean = this in

Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE

* Returns the Unicode general

category of this character.

@SinceKotlin("1.5")

public actual val Char.category:

CharCategory

get() = CharCategory.valueOf(getCategoryValue())

* Returns `true` if this character

(Unicode code point) is defined in Unicode.

* A character is considered to be defined in Unicode if its

[category] is not [CharCategory.UNASSIGNED].

@SinceKotlin("1.5")

public actual fun Char.isDefined():

Boolean {

 if (this < "\u0080") {

 return true

 }

 return getCategoryValue() !=

CharCategory.UNASSIGNED.value

}
* Returns `true` if this character is a letter.

* A character is

considered to be a letter if its [category] is [CharCategory.UPPERCASE_LETTER],

[CharCategory.LOWERCASE_LETTER], [CharCategory.TITLECASE_LETTER],

[CharCategory.MODIFIER_LETTER], or [CharCategory.OTHER_LETTER].

* @sample

samples.text.Chars.isLetter

@SinceKotlin("1.5")

public actual fun Char.isLetter(): Boolean {

 if (this in

'a'..'z' || this in 'A'..'Z') {

 return true

 }

 if (this < "\u0080") {

 return false

 }

 return isLetterImpl()

}
* Returns `true` if this character is a letter or digit.

* @see isLetter

* @see isDigit

* @sample samples.text.Chars.isLetterOrDigit

@SinceKotlin("1.5")

public actual fun

Char.isLetterOrDigit(): Boolean {

 if (this in 'a'..'z' || this in 'A'..'Z' || this in '0'..'9') {

 return true

 }

 if

(this < "\u0080") {

 return false

 }

 return isDigitImpl() || isLetterImpl()

}
* Returns `true` if

this character is a digit.

* A character is considered to be a digit if its [category] is

[CharCategory.DECIMAL_DIGIT_NUMBER].

* @sample samples.text.Chars.isDigit

@SinceKotlin("1.5")

public actual fun Char.isDigit(): Boolean {

 if (this in '0'..'9') {

 return true

 }

 if (this < "\u0080") {

 return false

 }

 return isDigitImpl()

}
* Returns `true` if this

character is upper case.

* A character is considered to

be an upper case character if its [category] is [CharCategory.UPPERCASE_LETTER],

* or it has contributory

property `Other_Uppercase` as defined by the Unicode Standard.

* @sample

samples.text.Chars.isUpperCase

@SinceKotlin("1.5")

public actual fun Char.isUpperCase(): Boolean {

 if (this in 'A'..'Z') {

 return true

 }

 if (this < "\u0080") {

 return false

 }

 return

isUpperCaseImpl()

}
* Returns `true` if this character is lower case.

* A character is considered to

be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],

* or it has contributory

property `Other_Lowercase` as defined by the Unicode Standard.

* @sample

samples.text.Chars.isLowerCase

@SinceKotlin("1.5")

public actual fun Char.isLowerCase(): Boolean {

 if (this in 'a'..'z') {

 return true

 }

 if (this < "\u0080") {

 return false

 }

 return

isLowerCaseImpl()

}
* Returns

`true` if this character is a title case letter.

* A character is considered to be a title case letter if its [category] is

[CharCategory.TITLECASE_LETTER].

* @sample samples.text.Chars.isTitleCase

@SinceKotlin("1.5")

public actual fun Char.isTitleCase(): Boolean {

 if (this < "\u0080") {

 return

false

 }

 return getCategoryValue() == CharCategory.TITLECASE_LETTER.value

}
* Returns

`true` if this character is an ISO control character.

* A character is considered to be an ISO control character if

```

```

its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range ``\u0000..\u001F`` or in the
range ``\u007F..\u009F``.\n * @sample samples.text.Chars.isISOControl\n * ^\n @SinceKotlin("1.5")\n public
actual fun Char.isISOControl(): Boolean {\n return this <= "\u001F" || this in "\u007F..\u009F"\n }\n\n **\n *
Determines whether a character is whitespace according to the Unicode standard.\n * Returns `true`
if the character is whitespace.\n * ^\n * @sample samples.text.Chars.isWhitespace\n * ^\n public actual fun
Char.isWhitespace(): Boolean = isWhitespaceImpl(),"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n * ^\n\n package kotlin.text\n\n import kotlin.js.RegExp\n\n /**\n * Converts
the characters in the specified array to a string.\n * ^\n @SinceKotlin("1.2")\n @Deprecated("Use
CharArray.concatToString() instead",
ReplaceWith("chars.concatToString()"))\n @DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5")\n public actual fun String(chars: CharArray): String {\n var result = ""\n for (char in chars) {\n
result += char\n }\n return result\n }\n\n **\n * Converts the characters from a portion of the specified array to a
string.\n * ^\n * @throws IndexOutOfBoundsException if either [offset] or [length]
are less than zero\n * or `offset + length` is out of [chars] array bounds.\n
* ^\n @SinceKotlin("1.2")\n @Deprecated("Use CharArray.concatToString(startIndex, endIndex) instead",
ReplaceWith("chars.concatToString(offset, offset + length)"))\n @DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5")\n public actual fun String(chars: CharArray, offset: Int, length: Int): String {\n if (offset < 0 ||
length < 0 || chars.size - offset < length)\n throw IndexOutOfBoundsException("size: ${chars.size}; offset:
$offset; length: $length")\n var result = ""\n for (index in offset until offset + length) {\n result +=
chars[index]\n }\n return result\n }\n\n **\n * Concatenates characters in this [CharArray] into a String.\n
* ^\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual fun
CharArray.concatToString(): String {\n var result = ""\n for (char in this) {\n result += char\n }\n
return result\n }\n\n **\n * Concatenates characters in this [CharArray] or its subrange into a String.\n * ^\n * @param startIndex the
beginning (inclusive) of the subrange of characters, 0 by default.\n * @param endIndex the end (exclusive) of the
subrange of characters, size of this array by default.\n * @throws IndexOutOfBoundsException if [startIndex] is
less than zero or [endIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[startIndex] is greater than [endIndex].\n
* ^\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun CharArray.concatToString(startIndex: Int = 0,
endIndex: Int = this.size): String {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n var
result = ""\n for (index in startIndex until endIndex) {\n result += this[index]\n }\n return
result\n }\n\n **\n * Returns a [CharArray] containing characters of this string.\n
* ^\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual fun
String.toCharArray(): CharArray {\n return CharArray(length) { get(it) }\n }\n\n **\n * Returns a [CharArray]
containing characters of this string or its substring.\n * ^\n * @param startIndex the beginning (inclusive) of the
substring, 0 by default.\n * @param endIndex the end (exclusive) of the substring, length of this string by default.\n
* ^\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length
of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n
* ^\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int
= this.length): CharArray {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n return
CharArray(endIndex - startIndex)
 { get(startIndex + it) }\n }\n\n **\n * Decodes a string from the bytes in UTF-8 encoding in this array.\n * ^\n *
Malformed byte sequences are replaced by the replacement char ``\uFFFF``.\n
* ^\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual fun
ByteArray.decodeToString(): String {\n return decodeUtf8(this, 0, size, false)\n }\n\n **\n * Decodes a string from

```



the bytes in UTF-8 encoding in this array or its subrange.  
 \* @param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.  
 \* @param endIndex the end (exclusive) of the subrange to decode, size of this array by default.  
 \* @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char `        `.  
 \* @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.  
 \* @throws IllegalArgumentException if [startIndex] is greater than [endIndex].  
 \* @throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTIONS_WITH_DEFAULT_ARGUMENTS")\npublic actual fun ByteArray.decodeToString(\n startIndex: Int = 0,\n endIndex: Int = this.size,\n throwOnInvalidSequence: Boolean = false\n): String {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n return decodeUtf8(this, startIndex, endIndex,\n throwOnInvalidSequence)\n}\n\n/**\n * Encodes this string to an array of bytes in UTF-8 encoding.\n * Any malformed char sequence is replaced by the replacement byte sequence.\n */\n*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun String.encodeToByteArray(): ByteArray {\n return encodeUtf8(this, 0, length, false)\n}\n\n/**\n * Encodes this string or its substring to an array of bytes in UTF-8 encoding.\n * @param startIndex the beginning (inclusive) of the substring to encode, 0 by default.\n * @param endIndex the end (exclusive) of the substring to encode, length of this string by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on malformed char sequence or replace.\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n * @throws CharacterCodingException if this string contains malformed char sequence and [throwOnInvalidSequence] is true.\n */\n*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTIONS_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.encodeToByteArray(\n startIndex: Int = 0,\n endIndex: Int = this.length,\n throwOnInvalidSequence: Boolean = false\n): ByteArray {\n AbstractList.checkBoundsIndexes(\n startIndex, endIndex, length)\n return encodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence)\n}\n\nReturns a copy of this string converted to upper case using the rules of the default locale.\n*\n@Deprecated("Use uppercase() instead.", ReplaceWith("uppercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String =\n asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * @sample samples.text.Strings.uppercase\n */\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String =\n asDynamic().toUpperCase()\n\nReturns a copy of this string converted to lower case using the rules of the default locale.\n*\n@Deprecated("Use lowercase() instead.", ReplaceWith("lowercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String =\n asDynamic().toLowerCase()\n\nReturns a copy of this string converted to lower case using Unicode mapping rules of the invariant locale.\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * @sample samples.text.Strings.lowercase\n */\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String =\n asDynamic().toLowerCase()\n\ninternal actual inline fun String.nativeIndexOf(str: String, fromIndex: Int): Int =\n asDynamic().indexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal

```



## Language

contributors.  
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CharsKit")\n\npackage kotlin.text\n\n/**\n
```

Returns the numeric value of the decimal digit that this Char represents. Throws an exception if this Char is not a valid decimal digit.  
A Char is considered to represent a decimal digit if [isDigit] is true for the Char.

In this case, the Unicode decimal digit value of the character is returned.  
@sample

```
samples.text.Chars.digitToInt\n
```

```
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToInt(): Int {\n return digitOf(this, 10).also { if (it < 0) throw IllegalArgumentException("Char $this is not a decimal digit") }\n}\n\n/**\n
```

Returns the numeric value of the digit that this Char represents in the specified [radix].  
Throws an exception if the [radix]

is not in the range `2..36` or if this Char is not a valid digit in the specified [radix].  
A Char is considered to represent a digit in the specified [radix] if at least one of the following is true:  
- [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit value is returned.  
- The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.  
- The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.  
- The Char is one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.  
- The Char is one of the fullwidth Latin small letters '\uFF41'

through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.  
@sample samples.text.Chars.digitToInt

```
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToInt(radix: Int): Int {\n return digitToIntOrNull(radix) ?: throw IllegalArgumentException("Char $this is not a digit in the given radix=$radix")\n}\n\n/**\n
```

Returns the numeric value of the decimal digit that this Char represents, or `null` if this Char is not a valid decimal digit.  
A Char is considered to represent a decimal digit if [isDigit] is true for the Char.  
In this case, the Unicode decimal digit value of the character is returned.  
@sample

```
samples.text.Chars.digitToIntOrNull\n
```

```
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToIntOrNull(): Int? {\n return digitOf(this, 10).takeIf { it >= 0 }\n}\n\n/**\n
```

Returns the numeric value of the digit that this Char represents in the specified [radix], or `null` if this Char is not a valid digit in the specified [radix].  
Throws an exception if the [radix] is not in the range `2..36`.  
A Char is considered to represent a digit in the specified [radix] if at least one of the following is true:  
- [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit value is returned.  
- The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.  
- The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.  
- The Char is one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`.

In this case, `this.code - 0xFF21 + 10` is returned.  
- The Char is one of the fullwidth Latin small letters '\uFF41' through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.  
@sample samples.text.Chars.digitToIntOrNull

```
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToIntOrNull(radix: Int): Int? {\n checkRadix(radix)\n return digitOf(this, radix).takeIf { it >= 0 }\n}\n\n/**\n
```

Returns the Char that represents this decimal digit.  
Throws an exception if this value is not in the range `0..9`.  
If this value is in `0..9`, the decimal digit Char with code `0'.code + this` is returned.  
@sample

```

@sample samples.text.Chars.digitToChar\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(): Char
{\n if (this in 0..9) {\n return '0' + this\n }\n throw IllegalArgumentException("\nInt $this
is not a decimal digit")\n}\n\n/**\n * Returns the Char that represents this numeric digit value in the specified
[radix].\n * Throws an exception if the [radix] is not in the range `2..36` or if this value is not in the range `0` until
radix`.\n * If this value is less than `10`, the decimal digit Char with code `0.code + this` is returned.\n *
Otherwise, the uppercase Latin letter with code `A.code + this - 10` is returned.\n *\n * @sample
samples.text.Chars.digitToChar\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(radix:
Int): Char {\n if (radix !in 2..36) {\n throw IllegalArgumentException("Invalid radix: $radix. Valid radix
values are in range 2..36")\n }\n if (this < 0 || this >= radix) {\n throw IllegalArgumentException("Digit
$this does not represent a valid digit in radix $radix")\n }\n return if (this < 10) {\n '0' + this\n } else {\n
'A' + this - 10\n }\n}\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n
*\n *\n@Deprecated("Use lowercaseChar() instead.")\n\nReplaceWith("lowercaseChar()")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
Char.toLowerCase(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself
is returned.\n *\n * @sample samples.text.Chars.lowercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.lowercaseChar(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.\n
*\n * For example, ``\u0130.toLowerCase()`` returns ``\u0069\u0307``,\n * where ``\u0130`` is the LATIN CAPITAL
LETTER I WITH DOT ABOVE character (``\u0130``).\n * If this character has no lower case mapping, the result of
`toString()` of this char is returned.\n *\n * @sample samples.text.Chars.lowercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.lowercase(): String\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n *\n@Deprecated("Use uppercaseChar() instead.")\n\nReplaceWith("uppercaseChar()")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
Char.toUpperCase(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character
itself is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.uppercaseChar(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.\n * For example, ``\uFB00.toUpperCase()`` returns ``\u0046\u0046``,\n * where
``\uFB00`` is the LATIN SMALL LIGATURE FF character (``\ufb00``).\n * If this character has no upper case
mapping, the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.uppercase(): String\n\n/**\n * Converts this character to title case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n *\n * To support one-to-many
character mapping use the [titlecase] function.\n * If this character has no mapping
equivalent, the result of calling [uppercaseChar] is returned.\n *\n * @sample samples.text.Chars.titlecase\n
*\n@SinceKotlin("1.5")\npublic expect fun Char.titlecaseChar(): Char\n\n/**\n * Converts this character to title
case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character

```

mapping, thus the length of the returned string can be greater than one.  
 For example, `'\uFB00'.titlecase()` returns `\"'\u0046\u0066\"'`, where `\"'\uFB00\"` is the LATIN SMALL LIGATURE FF character (`\"'\uFB00\"`).  
 If this character has no title case mapping, the result of `[uppercase]` is returned instead.  
 @sample  
`samples.text.Chars.titlecase`  
 @SinceKotlin("1.5")  
`public fun Char.titlecase(): String = titlecaseImpl()`  
 Concatenates this Char and a String.  
 @sample  
`samples.text.Chars.plus`  
 @kotlin.internal.InlineOnly  
`public inline operator fun Char.plus(other: String): String = this.toString() + other`  
 Returns `\"true\"` if this character is equal to the `[other]` character, optionally ignoring character case.  
 Two characters are considered equal ignoring case if `\"Char.uppercaseChar().lowercaseChar()\"` on each character produces the same result.  
 @param `ignoreCase` `\"true\"` to ignore character case when comparing characters. By default `\"false\"`.  
 @sample  
`samples.text.Chars.equals`  
`public fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {`  
 if (this == other) return true  
 if (!ignoreCase) return false  
 val thisUpper = this.uppercaseChar()  
 val otherUpper = other.uppercaseChar()  
 return thisUpper == otherUpper || thisUpper.lowercaseChar() == otherUpper.lowercaseChar()  
 }  
 Returns `\"true\"` if this character is a Unicode surrogate code unit.  
`public fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE`  
 Returns the Unicode general category of this character.  
 @SinceKotlin("1.5")  
`public expect val Char.category: CharCategory`  
 Returns `\"true\"` if this character (Unicode code point) is defined in Unicode.  
 A character is considered to be defined in Unicode if its `[category]` is not `[CharCategory.UNASSIGNED]`.  
 @SinceKotlin("1.5")  
`public expect fun Char.isDefined(): Boolean`  
 Returns `\"true\"` if this character is a letter.  
 A character is considered to be a letter if its `[category]` is `[CharCategory.UPPERCASE_LETTER]`, `[CharCategory.LOWERCASE_LETTER]`, `[CharCategory.TITLECASE_LETTER]`, `[CharCategory.MODIFIER_LETTER]`, or `[CharCategory.OTHER_LETTER]`.  
 @sample  
`samples.text.Chars.isLetter`  
 @SinceKotlin("1.5")  
`public expect fun Char.isLetter(): Boolean`  
 Returns `\"true\"` if this character is a letter or digit.  
 @see `isLetter`  
 @see `isDigit`  
 @sample  
`samples.text.Chars.isLetterOrDigit`  
 @SinceKotlin("1.5")  
`public expect fun Char.isLetterOrDigit(): Boolean`  
 Returns `\"true\"` if this character is a digit.  
 A character is considered to be a digit if its `[category]` is `[CharCategory.DECIMAL_DIGIT_NUMBER]`.  
 @sample  
`samples.text.Chars.isDigit`  
 @SinceKotlin("1.5")  
`public expect fun Char.isDigit(): Boolean`  
 Returns `\"true\"` if this character is upper case.  
 A character is considered to be an upper case character if its `[category]` is `[CharCategory.UPPERCASE_LETTER]`, or it has contributory property `\"Other_Uppercase\"` as defined by the Unicode Standard.  
 @sample  
`samples.text.Chars.isUpperCase`  
 @SinceKotlin("1.5")  
`public expect fun Char.isUpperCase(): Boolean`  
 Returns `\"true\"` if this character is lower case.  
 A character is considered to be a lower case character if its `[category]` is `[CharCategory.LOWERCASE_LETTER]`, or it has contributory property `\"Other_Lowercase\"` as defined by the Unicode Standard.  
 @sample  
`samples.text.Chars.isLowerCase`  
 @SinceKotlin("1.5")  
`public expect fun Char.isLowerCase(): Boolean`  
 Returns `\"true\"` if this character is a title case letter.  
 A character is considered to be a title case letter if its `[category]` is `[CharCategory.TITLECASE_LETTER]`.  
 @sample  
`samples.text.Chars.isTitleCase`  
 @SinceKotlin("1.5")  
`public expect fun Char.isTitleCase(): Boolean`  
 Returns `\"true\"` if this character is an ISO control character.  
 A character is considered to be an ISO control character if its `[category]` is `[CharCategory.CONTROL]`, meaning the Char is in the range `\"'\u0000'..\"'\u001F\"` or in the range `\"'\u007F'..\"'\u009F\"`.  
 @sample  
`samples.text.Chars.isISOControl`  
 @SinceKotlin("1.5")  
`public expect fun Char.isISOControl(): Boolean`  
 Determines whether a character is whitespace according to the Unicode standard.  
 Returns `\"true\"` if the character is whitespace.  
 @sample  
`samples.text.Chars.isWhitespace`  
 @public  
`expect fun Char.isWhitespace(): Boolean`  
 Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.  
 Use of this source code is governed by the Apache 2.0 license that can be

found in the license/LICENSE.txt file.\n \*^\\n\\npackage kotlin\\n\\n\\n/\*\*\n \* Creates a Char with the specified [code], or throws an exception if the [code] is out of `Char.MIN\_VALUE.code..Char.MAX\_VALUE.code`.\\n \*\\n \* If the program that calls this function is written in a way that only valid [code] is passed as the argument,\\n \* using the overload that takes a [UShort] argument is preferable (`Char(intValue.toUShort())`).\\n \* That overload doesn't check validity of the argument, and may improve program performance when the function is called routinely inside a loop.\\n \*\\n \* @sample samples.text.Chars.charFromCode\\n

```
*\\n@SinceKotlin("1.5")\\n@WasExperimental(ExperimentalStdlibApi::class)\\n@kotlin.internal.InlineOnly\\npublic inline fun Char(code:
```

```
Int): Char {\\n if (code < Char.MIN_VALUE.code || code > Char.MAX_VALUE.code) {\\n throw\n IllegalArgumentException("Invalid Char code: $code")\\n }\\n return code.toChar()\\n}\\n\\n/**\n * Creates a Char with the specified [code].\\n *\\n * @sample samples.text.Chars.charFromCode\\n
```

```
*\\n@SinceKotlin("1.5")\\n@WasExperimental(ExperimentalStdlibApi::class)\\n@Suppress("NO_ACTUAL_FOR_EXPECT")\\npublic expect fun Char(code: UShort): Char\\n\\n/**\n * Returns the code of this Char.\\n *\\n * Code of a Char is the value it was constructed with, and the UTF-16 code unit corresponding to this Char.\\n *\\n * @sample\n samples.text.Chars.code\\n
```

```
\\n@SinceKotlin("1.5")\\n@WasExperimental(ExperimentalStdlibApi::class)\\n@kotlin.internal.InlineOnly\\n@Suppress("DEPRECATION")\\n@kotlin.internal.IntrinsicConstEvaluation\\npublic inline val Char.code: Int get() =\n this.toInt()\\n", "/\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\\n * Use\n of this source code
```

is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\\n

```
\\n@file:kotlin.jvm.JvmMultifileClass\\n@file:kotlin.jvm.JvmName("SequencesKt")\\n\\npackage\n kotlin.sequences\\n\\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\\n// See:\n https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\\n\\nimport kotlin.random.\\n\\n/**\n * Returns\n `true` if [element] is found in the sequence.\\n *\\n * The operation is _terminal_.\\n *\\n@public operator fun\n <@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {\\n return indexOf(element)\n >= 0\\n}\\n\\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the\n [index] is out of bounds of this sequence.\\n *\\n * The operation is _terminal_.\\n *\\n * @sample\n samples.collections.Collections.Elements.elementAt\\n *\\n@public fun <T> Sequence<T>.elementAt(index: Int): T\n {\\n return elementAtOrElse(index) { throw IndexOutOfBoundsException("Sequence\n doesn't contain element at index $index.") }\\n}\\n\\n/**\n * Returns an element at the given [index] or the result of\n calling the [defaultValue] function if the [index] is out of bounds of this sequence.\\n *\\n * The operation is\n _terminal_.\\n *\\n * @sample\n samples.collections.Collections.Elements.elementAtOrElse\\n *\\n@public fun <T>\n Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\\n if (index < 0)\\n return\n defaultValue(index)\\n val iterator = iterator()\\n var count = 0\\n while (iterator.hasNext()) {\\n val element\n = iterator.next()\\n if (index == count++)\\n return element\\n }\\n return\n defaultValue(index)\\n}\\n\\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of\n this sequence.\\n *\\n * The operation is _terminal_.\\n *\\n * @sample\n samples.collections.Collections.Elements.elementAtOrNull\\n *\\n@public fun <T>\n Sequence<T>.elementAtOrNull(index: Int): T? {\\n if (index
```

```
< 0)\\n return null\\n val iterator = iterator()\\n var count = 0\\n while (iterator.hasNext()) {\\n val\n element = iterator.next()\\n if (index == count++)\\n return element\\n }\\n return null\\n}\\n\\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\\n *\\n * The\n operation is _terminal_.\\n *\\n * @sample\n samples.collections.Collections.Elements.find\\n
```

```
*\\n@kotlin.internal.InlineOnly\\npublic inline fun <T> Sequence<T>.find(predicate: (T) -> Boolean): T? {\\n\n return firstOrNull(predicate)\\n}\\n\\n/**\n * Returns the last element matching the given [predicate], or `null` if no\n such element was found.\\n *\\n * The operation is _terminal_.\\n *\\n * @sample\n samples.collections.Collections.Elements.find\\n *\\n@kotlin.internal.InlineOnly\\npublic inline fun <T>\n Sequence<T>.findLast(predicate: (T) -> Boolean): T? {\\n return lastOrNull(predicate)\\n}\\n\\n/**\n * Returns the
```

```

first element.\n *\n * The operation
is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\npublic fun <T>
Sequence<T>.first(): T {\n val iterator = iterator()\n if (!iterator.hasNext())\n throw
NoSuchElementException("Sequence is empty.")\n return iterator.next()\n}\n\n/**\n * Returns the first element
matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n *\n * The
operation is _terminal_.\n *\npublic inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T {\n for
(element in this) if (predicate(element)) return element\n throw NoSuchElementException("Sequence contains no
element matching the predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function
being applied to elements of this sequence in iteration order,\n * or throws [NoSuchElementException] if no non-
null value was produced.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the sequence was transformed to a non-null value.")\n}\n\n/**\n *
Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or `null` if no non-null value was produced.\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this) {\n val result =
transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n *
Returns the first element, or `null` if the
sequence is empty.\n *\n * The operation is _terminal_.\n *\npublic fun <T> Sequence<T>.firstOrNull(): T? {\n
val iterator = iterator()\n if (!iterator.hasNext())\n return null\n return iterator.next()\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if element was not found.\n *\n * The operation
is _terminal_.\n *\npublic inline fun <T> Sequence<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element
in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns first index of [element], or -1
if the sequence does not contain element.\n *\n * The operation is _terminal_.\n *\npublic fun
<@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {\n var index = 0\n for (item in
this) {\n checkIndexOverflow(index)\n if (element == item)\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given
[predicate], or -1 if the sequence does not contain such element.\n *\n * The operation is _terminal_.\n *\npublic
inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n var index = 0\n for (item in this)
{\n checkIndexOverflow(index)\n if (predicate(item))\n return index\n index++\n }\n return
-1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the sequence does not
contain such element.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T>
Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n var lastIndex = -1\n var index = 0\n for (item
in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n lastIndex = index\n index++\n
 }\n return lastIndex\n}\n\n/**\n * Returns the last element.\n *\n * The operation is _terminal_.\n *\n * @throws
NoSuchElementException if the sequence is empty.\n *\n * @sample
samples.collections.Collections.Elements.last\n
*\npublic fun <T> Sequence<T>.last(): T {\n val iterator = iterator()\n if (!iterator.hasNext())\n throw
NoSuchElementException("Sequence is empty.")\n var last = iterator.next()\n while (iterator.hasNext())\n
last = iterator.next()\n return last\n}\n\n/**\n * Returns the last element matching the given [predicate].\n *\n *
The operation is _terminal_.\n *\n * @throws NoSuchElementException if no such element is found.\n *\n * @sample
samples.collections.Collections.Elements.last\n *\npublic inline fun <T> Sequence<T>.last(predicate: (T)
-> Boolean): T {\n var last: T? = null\n var found = false\n for (element in this) {\n if (predicate(element))
{\n last = element\n found = true\n }\n }\n if (!found) throw
NoSuchElementException("Sequence contains no element matching the predicate.")\n}

```

```

@Suppress("UNCHECKED_CAST")\n return last as T\n}\n\n/**\n * Returns
last index of [element], or -1 if the sequence does not contain element.\n *\n * The operation is _terminal_.\n
*/\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {\n var lastIndex
= -1\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (element == item)\n lastIndex = index\n index++\n }\n return lastIndex\n}\n\n/**\n * Returns the last element, or `null` if the
sequence is empty.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Sequence<T>.lastOrNull(): T? {\n val iterator
= iterator()\n if (!iterator.hasNext())\n return null\n var last = iterator.next()\n while (iterator.hasNext())\n last = iterator.next()\n return last\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null`
if no such element was found.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T>
Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n var last: T? = null\n for (element in this) {\n if
(predicate(element)) {\n last = element\n }\n }\n return last\n}\n\n/**\n * Returns the single element,
or throws an exception if the sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n
*/\npublic fun <T> Sequence<T>.single(): T {\n val iterator = iterator()\n if (!iterator.hasNext())\n throw
NoSuchElementException("Sequence is empty.")\n val single = iterator.next()\n if (iterator.hasNext())\n throw
IllegalArgumentException("Sequence has more than one element.")\n return single\n}\n\n/**\n * Returns
the single element matching the given [predicate], or throws exception if there is no or more than one matching
element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.single(predicate:
(T) -> Boolean): T {\n var single: T? = null\n var found = false\n for (element in this) {\n if
(predicate(element)) {\n if (found) throw IllegalArgumentException("Sequence contains more than one
matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw
NoSuchElementException("Sequence contains no element matching the predicate.")\n}\n\n@Suppress("UNCHECKED_CAST")\n return single as T\n}\n\n/**\n * Returns single element, or `null` if the
sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n */\npublic fun <T>
Sequence<T>.singleOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n val
single = iterator.next()\n if (iterator.hasNext())\n return null\n return single\n}\n\n/**\n * Returns the single
element matching the given [predicate], or `null` if element was not found or more than
one element was found.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T>
Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n var single: T? = null\n var found = false\n for
(element in this) {\n if (predicate(element)) {\n if (found) return null\n single = element\n
found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a sequence containing
all elements except first [n] elements.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Sequence<T>.drop(n: Int):
Sequence<T> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n return when {\n n
== 0 -> this\n this is DropTakeSequence -> this.drop(n)\n else -> DropSequence(this, n)\n }\n}\n\n/**\n * Returns a sequence containing
all elements except first elements that satisfy the given [predicate].\n *\n * The operation is _intermediate_ and
stateless.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun <T>
Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> {\n return DropWhileSequence(this,
predicate)\n}\n\n/**\n * Returns a sequence containing only elements matching the given [predicate].\n *\n * The
operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {\n return
FilteringSequence(this, true, predicate)\n}\n\n/**\n * Returns a sequence containing only elements matching the
given [predicate].\n *\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n *\n * The operation is _intermediate_ and _stateless_.\n
*/\n\n

```



```

@sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic fun <T>
Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> {\n // TODO: Rewrite with
generalized MapFilterIndexingSequence\n return
TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) }), { it.value
})\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n *\npublic inline fun <T, C : MutableCollection<in T>>
Sequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n forEachIndexed {
index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return
destination\n}\n\n/**\n
* Returns a sequence containing all elements that are instances of specified type parameter R.\n *\n * The operation
is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Collections.Filtering.filterIsInstance\n
\npublic inline fun <reified R> Sequence<>.filterIsInstance(): Sequence<@kotlin.internal.NoInfer R> {\n
@Suppress(\"UNCHECKED_CAST\")\n return filter { it is R } as Sequence<R>\n}\n\n/**\n * Appends all
elements that are instances of specified type parameter R to the given [destination].\n *\n * The operation is
intermediate.\n *\n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n *\npublic inline fun
<reified R, C : MutableCollection<in R>> Sequence<*>.filterIsInstanceTo(destination: C): C {\n for (element in
this) if (element is R) destination.add(element)\n return destination\n}\n\n/**\n * Returns a sequence containing
all elements not matching the given [predicate].\n *\n * The operation is _intermediate_
and _stateless_.\n *\n * @sample samples.collections.Collections.Filtering.filter\n *\npublic fun <T>
Sequence<T>.filterNot(predicate: (T) -> Boolean): Sequence<T> {\n return FilteringSequence(this, false,
predicate)\n}\n\n/**\n * Returns a sequence containing all elements that are not `null`.\n *\n * The operation is
intermediate and _stateless_.\n *\n * @sample samples.collections.Collections.Filtering.filterNotNull\n
*\npublic fun <T : Any> Sequence<T?>.filterNotNull(): Sequence<T> {\n
@Suppress(\"UNCHECKED_CAST\")\n return filterNot { it == null } as Sequence<T>\n}\n\n/**\n * Appends all
elements that are not `null` to the given [destination].\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C : MutableCollection<in T>, T : Any>
Sequence<T?>.filterNotNullTo(destination: C): C {\n for (element in this) if (element != null)
destination.add(element)\n return destination\n}\n\n/**\n
* Appends all elements not matching the given [predicate] to the given [destination].\n *\n * The operation is
intermediate.\n *\n * @sample samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C :
MutableCollection<in T>> Sequence<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in
T>> Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if
(predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Returns a sequence containing
first [n] elements.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @throws
IllegalArgumentException
if [n] is negative.\n *\n * @sample samples.collections.Collections.Transformations.take\n *\npublic fun <T>
Sequence<T>.take(n: Int): Sequence<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\"
}\n return when {\n n == 0 -> emptySequence()\n this is DropTakeSequence -> this.take(n)\n else ->
TakeSequence(this, n)\n }\n}\n\n/**\n * Returns a sequence containing first elements satisfying the given
[predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun <T> Sequence<T>.takeWhile(predicate: (T) -
> Boolean): Sequence<T> {\n return TakeWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence that
yields elements of this sequence sorted according to their natural sort order.\n *\n * The sort is _stable_. It means

```

that equal elements preserve their order relative to each other after sorting.

```

 * The operation is
 intermediate and _stateful_.
 public fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {
 return object : Sequence<T> {
 override fun iterator(): Iterator<T> {
 val sortedList =
 this@sorted.toMutableList()
 sortedList.sort()
 return sortedList.iterator()
 }
 }
}

```

\* Returns a sequence that yields elements of this sequence sorted according to natural sort order of the value returned by specified [selector] function.

\* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

\* The operation is `_intermediate_ and _stateful_`.

\* @sample samples.collections.Collections.Sorting.sortedBy

```

 public inline fun <T, R : Comparable<R>>
 Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {
 return
 sortedWith(compareBy(selector))
}

```

\* Returns a sequence that yields elements of this sequence sorted descending according to natural sort order of the value returned by specified [selector] function.

\* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

\* The operation is `_intermediate_ and _stateful_`.

```

 public inline fun <T, R : Comparable<R>> Sequence<T>.sortedByDescending(crossinline selector:
 (T) -> R?): Sequence<T> {
 return sortedWith(compareByDescending(selector))
}

```

\* Returns a sequence that yields elements of this sequence sorted descending according to their natural sort order.

\* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

\* The operation is `_intermediate_ and _stateful_`.

```

 public fun <T : Comparable<T>>
 Sequence<T>.sortedDescending(): Sequence<T> {
 return sortedWith(reverseOrder())
}

```

\* Returns a sequence that yields elements of this sequence sorted according to the specified [comparator].

\* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.

\* The operation is `_intermediate_ and _stateful_`.

```

 public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>):
 Sequence<T> {
 return object : Sequence<T> {
 override fun iterator(): Iterator<T> {
 val
 sortedList = this@sortedWith.toMutableList()
 sortedList.sortWith(comparator)
 return
 sortedList.iterator()
 }
 }
}

```

\* Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given sequence.

\* If any of two pairs would have the same key the last one gets added to the map.

\* The returned map preserves the entry iteration order of the original sequence.

\* The operation is `_terminal_`.

\* @sample samples.collections.Collections.Transformations.associate

```

 public inline fun <T, K, V> Sequence<T>.associate(transform: (T) -> Pair<K,
 V>): Map<K, V> {
 return associateTo(LinkedHashMap<K, V>(), transform)
}

```

\* Returns a [Map] containing the elements from the given sequence indexed by the key returned from [keySelector] function applied to each element.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

\* The returned map preserves the entry iteration order of the original sequence.

\* The operation is `_terminal_`.

\* @sample samples.collections.Collections.Transformations.associateBy

```

 public inline fun <T, K> Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> {
 return
 associateByTo(LinkedHashMap<K, T>(), keySelector)
}

```

\* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given sequence.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

\* The returned map preserves the entry iteration order of the original sequence.

\* The operation is `_terminal_`.

\* @sample samples.collections.Collections.Transformations.associateByWithValueTransform

```

 public inline
 fun <T, K, V> Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
 return
 associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)
}

```

\* Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given sequence and value is the element itself.

\* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

\* The operation is `_terminal_`.

\* @sample samples.collections.Collections.Transformations.associateByTo

```

 public inline fun <T, K, M :
 MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M {
 for

```

(element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/\*\*\n \* Populates and returns the [destination] mutable map with key-value pairs,\n \* where key is provided by the [keySelector] function and\n \* and value is provided by the [valueTransform] function applied to elements of the given sequence.\n \* \n \* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n \* \n \* The operation is \_terminal\_.\n \* \n \* @sample samples.collections.Collections.Transformations.associateByToWithValueTransform\n \*/\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/\*\*\n \* Populates and returns the [destination] mutable map with key-value pairs\n \* provided by [transform] function applied to each element of the given sequence.\n \* \n \* If any of two pairs would have the same key the last one gets added to the map.\n \* \n \* The operation is \_terminal\_.\n \* \n \* @sample samples.collections.Collections.Transformations.associateTo\n \*/\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/\*\*\n \* Returns a [Map] where keys are elements from the given sequence and values are\n \* produced by the [valueSelector] function applied to each element.\n \* \n \* If any two elements are equal, the last one gets added to the map.\n \* \n \* The returned map preserves the entry iteration order of the original sequence.\n \* \n \* The operation is \_terminal\_.\n \* \n \* @sample samples.collections.Collections.Transformations.associateWith\n \*/\n@SinceKotlin("1.3")\npublic inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n return associateWithTo(result, valueSelector)\n}\n\n/\*\*\n \* Populates and returns the [destination] mutable map with key-value pairs for each element of the given sequence,\n \* where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n \* \n \* If any two elements are equal, the last one overwrites the former value in the map.\n \* \n \* The operation is \_terminal\_.\n \* \n \* @sample samples.collections.Collections.Transformations.associateWithTo\n \*/\n@SinceKotlin("1.3")\npublic inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/\*\*\n \* Appends all elements to the given [destination] collection.\n \* \n \* The operation is \_terminal\_.\n \*/\npublic fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/\*\*\n \* Returns a new [HashSet] of all elements.\n \* \n \* The operation is \_terminal\_.\n \*/\npublic fun <T> Sequence<T>.toHashSet(): HashSet<T> {\n return toCollection(HashSet<T>())\n}\n\n/\*\*\n \* Returns a [List] containing all elements.\n \* \n \* The operation is \_terminal\_.\n \*/\npublic fun <T> Sequence<T>.toList(): List<T> {\n return this.toMutableList().optimizeReadOnlyList()\n}\n\n/\*\*\n \* Returns a new [MutableList] filled with all elements of this sequence.\n \* \n \* The operation is \_terminal\_.\n \*/\npublic fun <T> Sequence<T>.toMutableList(): MutableList<T> {\n return toCollection(ArrayList<T>())\n}\n\n/\*\*\n \* Returns a [Set] of all elements.\n \* \n \* The returned set preserves the element iteration order of the original sequence.\n \* \n \* The operation is \_terminal\_.\n \*/\npublic fun <T> Sequence<T>.toSet(): Set<T> {\n return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()\n}\n\n/\*\*\n \* Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.\n \* \n \* The operation is \_intermediate\_ and \_stateless\_.\n \* \n \* @sample samples.collections.Collections.Transformations.flatMap\n \*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIterable")\npublic fun <T, R> Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {\n return FlatteningSequence(this, transform, Iterable<R>::iterator)\n}\n\n/\*\*\n \* Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.\n \* \n \* The operation is \_intermediate\_ and

```

stateless.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n * \npublic fun <T, R>
Sequence<T>.flatMap(transform:
(T) -> Sequence<R>): Sequence<R> {\n return FlatteningSequence(this, transform,
Sequence<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements yielded from results of [transform]
function being invoked on each element\n * and its index in the original sequence.\n * \n * The operation is
intermediate and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\npublic fun <T, R>
Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): Sequence<R> {\n return
flatMapIndexed(this, transform, Iterable<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements
yielded from results of [transform] function being invoked on each element\n * and its index in the original
sequence.\n * \n * The operation is _intermediate_
and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\npublic fun <T, R>
Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): Sequence<R> {\n return
flatMapIndexed(this, transform, Sequence<R>::iterator)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original sequence, to the given
[destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index:
Int, T) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(checkIndexOverflow(index++), element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original sequence, to the given [destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(checkIndexOverflow(index++), element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given
[destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIterableTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n * \n * The operation is _terminal_.\n * \npublic inline fun
<T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C
{\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original sequence by
the key returned by the given [keySelector] function\n * applied to each element and returns a map where each
group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration
order of the keys produced from the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T, K>
Sequence<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to
each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The
returned
 map preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The operation is
terminal.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\npublic inline fun <T, K, V> Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K,
List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups elements of the original sequence by the key returned by the given
[keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with
a list of corresponding elements.\n * \n * @return The [destination] map.\n *\n * The operation is _terminal_.\n * \n
* @sample samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T, K, M :
MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n
for (element in this) {\n val key = keySelector(element)\n
 val list = destination.getOrPut(key) { ArrayList<T>() } list.add(element)\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original sequence\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to
the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The
[destination] map.\n *\n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() } list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Creates a [Grouping]
 source from a sequence to be used later with one of group-and-fold operations\n * using the specified [keySelector]
function to extract a key from each element.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n *
@sample samples.collections.Grouping.groupingByEachCount\n *\n@SinceKotlin("1.1")\npublic inline fun <T,
K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n return object :
Grouping<T, K> {\n override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n override fun
keyOf(element: T): K = keySelector(element)\n }\n}\n\n/**\n * Returns a sequence containing the results of
applying the given [transform] function\n * to each element in the original sequence.\n *\n * The operation is
intermediate and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.map\n *\npublic
fun <T, R> Sequence<T>.map(transform: (T) -> R): Sequence<R> {\n return TransformingSequence(this,
transform)\n}\n\n/**\n * Returns a sequence containing the results of applying the given [transform] function\n * to
each element and its index in the original sequence.\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n *\n * The
operation is _intermediate_ and _stateless_.\n *\npublic fun <T, R> Sequence<T>.mapIndexed(transform: (index:
Int, T) -> R): Sequence<R> {\n return TransformingIndexedSequence(this, transform)\n}\n\n/**\n * Returns a
sequence containing only the non-null results of applying the given [transform] function\n * to each element and its
index in the original sequence.\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n *\n * The operation is _intermediate_
and _stateless_.\n *\npublic fun <T, R : Any> Sequence<T>.mapIndexedNotNull(transform: (index:
Int, T) -> R?): Sequence<R> {\n return TransformingIndexedSequence(this,
transform).filterNotNull()\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the
original sequence\n * and appends only the non-null results to the given [destination].\n * @param [transform]
function that takes the index of an element and the element itself\n * and returns the result of the transform applied

```

to the element.

```

 * The operation is _terminal_.
 * public inline fun <T, R : Any, C : MutableCollection<in R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
 forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }
 return destination
 }
 * Applies the given [transform] function to each element and its index in the original sequence
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element
 itself
 * and returns the result of the transform applied to the element.
 * The operation is _terminal_.
 * public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.mapIndexedTo(destination: C, transform:
 (index: Int, T) -> R): C {
 var index = 0
 for (item in this)
 destination.add(transform(checkIndexOverflow(index++), item))
 return destination
 }
 * Returns a sequence containing only the non-null results of applying the given [transform] function
 * to each element in the original sequence.
 * The operation is _intermediate_ and _stateless_.
 * @sample
 samples.collections.Collections.Transformations.mapNotNull
 * public fun <T, R : Any> Sequence<T>.mapNotNull(transform: (T) -> R?): Sequence<R> {
 return TransformingSequence(this, transform).filterNotNull()
 }
 * Applies the given [transform] function to each element in the original
 sequence
 * and appends only the non-null results to the given [destination].
 * The operation is _terminal_.
 * public inline fun <T, R : Any, C : MutableCollection<in R>> Sequence<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
 forEach { element ->
 transform(element)?.let { destination.add(it) } }
 return destination
 }
 * Applies the given [transform]
 function to each element of the original sequence
 * and appends the results to the given [destination].
 * The operation is _terminal_.
 * public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.mapTo(destination: C, transform: (T) -> R): C {
 for (item in this)
 destination.add(transform(item))
 return destination
 }
 * Returns a sequence that wraps each element of
 the original sequence
 * into an [IndexedValue] containing the index of that element and the element itself.
 * The operation is _intermediate_ and _stateless_.
 * public fun <T> Sequence<T>.withIndex():
 Sequence<IndexedValue<T>> {
 return IndexingSequence(this)
 }
 * Returns a sequence containing only distinct elements from the given sequence.
 * Among equal elements of
 the given sequence, only the first one will be present in the resulting sequence.
 * The elements in the resulting
 sequence are in the same order as they were in the source sequence.
 * The operation is _intermediate_ and
 stateful.
 * @sample
 samples.collections.Collections.Transformations.distinctAndDistinctBy
 * public fun <T> Sequence<T>.distinct(): Sequence<T> {
 return this.distinctBy { it }
 }
 * Returns a sequence
 containing only elements from the given sequence
 * having distinct keys returned by the given [selector]
 function.
 * Among elements of the given sequence with equal keys, only the first one will be present in the
 resulting sequence.
 * The elements in the resulting sequence are in the same order as they were in the source
 sequence.
 * The operation is _intermediate_ and _stateful_.
 * @sample
 samples.collections.Collections.Transformations.distinctAndDistinctBy
 * public fun <T, K> Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {
 return DistinctSequence(this, selector)
 }
 * Returns a new [MutableSet] containing all distinct elements from the given sequence.
 * The returned set
 preserves the element iteration order of the original sequence.
 * The operation is _terminal_.
 * public fun <T> Sequence<T>.toMutableSet(): MutableSet<T> {
 val set = LinkedHashSet<T>()
 for (item in this)
 set.add(item)
 return set
 }
 * Returns `true` if all elements match the given [predicate].
 * Note
 that if the sequence contains no elements, the function returns `true`
 * because there are no elements in it that _do
 not_ match the predicate.
 * See a more detailed explanation of this logic concept in ["Vacuous
 truth"](https://en.wikipedia.org/wiki/Vacuous_truth) article.
 * The operation is _terminal_.
 * @sample
 samples.collections.Collections.Aggregates.all
 * public inline fun <T> Sequence<T>.all(predicate: (T) ->
 Boolean): Boolean {
 for (element in this) if (!predicate(element)) return false
 return true
 }
 * Returns `true` if sequence has at least one element.
 * The operation is _terminal_.
 * @sample
 samples.collections.Collections.Aggregates.any
 * public fun <T> Sequence<T>.any(): Boolean {
 return

```

```

iterator().hasNext()\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n *\npublic inline fun <T> Sequence<T>.any(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns the number of elements in this sequence.\n *\n * The operation is _terminal_.\n *\npublic fun <T> Sequence<T>.count(): Int {\n var count = 0\n for (element in this) checkCountOverflow(++count)\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n *\n * The operation is _terminal_.\n *\npublic inline fun <T> Sequence<T>.count(predicate: (T) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) checkCountOverflow(++count)\n return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n *\n * Returns the specified [initial] value if the sequence is empty.\n *\n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T, R> Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original sequence.\n *\n * Returns the specified [initial] value if the sequence is empty.\n *\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T, R> Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n *\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n *\n * The operation is _terminal_.\n *\npublic inline fun <T> Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n var index = 0\n for (item in this) action(checkIndexOverflow(index++), item)\n}\n\n/**\n * Returns the largest element.\n *\n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\n * @since Kotlin("1.7")\n * @kotlin.jvm.JvmName("maxOrThrow")\n * @Suppress("CONFLICTING_OVERLOADS")\npublic fun Sequence<Double>.max(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n *\n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\n * @since Kotlin("1.7")\n * @kotlin.jvm.JvmName("maxOrThrow")\n * @Suppress("CONFLICTING_OVERLOADS")\npublic fun Sequence<Float>.max(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\n * @since Kotlin("1.7")\n * @kotlin.jvm.JvmName("maxOrThrow")\n * @Suppress("CONFLICTING_OVERLOADS")\npublic fun <T : Comparable<T>> Sequence<T>.max(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\n * @sample samples.collections.Collections.Aggregates.maxBy\n *\n * @since Kotlin("1.7")\n * @kotlin.jvm.JvmName("maxByOrThrow")\n * @Suppress("CONFLICTING_OVERL

```

```

OADS")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxBy(selector: (T) -> R): T {\n val iterator
= iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxElem = iterator.next()\n if
(!iterator.hasNext()) return maxElem\n var maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n
 val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while
(iterator.hasNext())\n

```

return maxElem}\n\n/\*\*\n \* Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n \* The operation is \_terminal\_.\n \* @sample

```

samples.collections.Collections.Aggregates.maxByOrNull\n *@\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.maxByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var
maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue <
v) {\n maxElem = e\n maxValue = v\n }\n } while (iterator.hasNext())\n return
maxElem}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the sequence.\n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n * The operation is _terminal_.\n * @throws NoSuchElementException if the sequence is
empty.\n

```

\*@\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) ->
Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/\*\*\n \* Returns the largest value among all
values produced by [selector] function\n \* applied to each element in the sequence.\n \* If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n \* The operation is \_terminal\_.\n \*
@throws NoSuchElementException if the sequence is empty.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.maxOf(selector: (T) -> Float): Float {\n val iterator = iterator()\n if
(!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the sequence.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * The operation is _terminal_.\n *
@throws NoSuchElementException if the sequence is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.maxOf(selector: (T) -> Double): Double? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n
 val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the sequence or `null` if there are no elements.\n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * The operation is _terminal_.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOf(selector: (T) -> R): R {\n val iterator = iterator()\n
 if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n if (maxValue < v) {\n maxValue = v\n
 }\n }\n return maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each element in the sequence or `null` if there are no elements.\n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * The operation is _terminal_.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no
elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * The
operation is _terminal_.\n

```

operation is \_terminal\_.\n



```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOrNull(selector:
(T) -> Float): Float? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max =
selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n max =
maxOf(max, v)\n }\n return max\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each

```

element in the sequence or `null` if there are no elements.\n \* The operation is \_terminal\_.

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOrNull(selector: (T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return
null\n var max = selector(iterator.next())\n while (iterator.hasNext()) {\n val v =
selector(iterator.next())\n if (max < v) {\n max = v\n }\n }\n return
max\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the sequence.\n * \n * @throws

```

NoSuchElementException if the sequence is empty.\n \* The operation is \_terminal\_.

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun <T, R> Sequence<T>.maxWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n val
iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max =
selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if
(comparator.compare(max, v) < 0) {\n max = v\n }\n }\n return max\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the sequence or `null` if there are no elements.\n * The operation is
terminal.

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>

```

```

Sequence<T>.maxWithOrNull(comparator: Comparator<in R>, selector:
(T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max =
selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if
(comparator.compare(max, v) < 0) {\n max = v\n }\n }\n return max\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *
The operation is _terminal_.

```

```

*\n@SinceKotlin("1.4")\npublic fun Sequence<Double>.maxOrNull():
Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *
The operation is _terminal_.

```

```

*\n@SinceKotlin("1.4")\npublic
fun Sequence<Float>.maxOrNull(): Float? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n * The operation

```

```

is _terminal_.
*\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Sequence<T>.maxOrNull(): T? {\n val
iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n *
Returns the first element having the largest value according to the provided [comparator].\n * The operation is
terminal.\n * \n * @throws NoSuchElementException if the sequence is empty.

```

```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic

```

```

fun <T> Sequence<T>.maxWith(comparator: Comparator<in T>): T {\n val iterator = iterator()\n if
(!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext())

```

```

{\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.maxOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n *\n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun Sequence<Double>.min(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n *\n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun Sequence<Float>.min(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun <T : Comparable<T>> Sequence<T>.min(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n *\n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minBy(selector: (T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n *\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) -> Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val

```

```

 v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n *
\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is
terminal.\n * \n * @throws NoSuchElementException if the sequence is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * The operation is
terminal.\n * \n * @throws NoSuchElementException if the sequence is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw
NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v
= selector(iterator.next())\n if (minValue > v) {\n minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is
`NaN`.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOfOrNull(selector:
(T) -> Double): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minValue =
selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is
terminal.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n
val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
sequence or `null` if there are no elements.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOfOrNull(selector: (T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return
null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v =
selector(iterator.next())\n if (minValue > v) {\n minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the
smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied
to each element in the sequence.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * The
operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n val iterator = iterator()\n if
(!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n

```

```

 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest
value according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n val iterator =
iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n
 minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are
no elements.\n *\n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\npublic
fun Sequence<Double>.minOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n
 var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n
 }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n *\n * If any of
elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n *\n@SinceKotlin("1.4")\npublic fun
Sequence<Float>.minOrNull(): Float? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n
 }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n *\n * The operation is
terminal.\n *\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Sequence<T>.minOrNull(): T? {\n
 val iterator = iterator()\n if (!iterator.hasNext())
 return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n
 if (min >
e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the
provided [comparator].\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the
sequence is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T {\n
 val iterator =
iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return
min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or
`null` if there are no elements.\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n
 val iterator = iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return
min\n}\n\n/**\n * Returns `true` if the sequence has no elements.\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Aggregates.none\n *\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.none(): Boolean {\n
 return !iterator().hasNext()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n *\n * The
operation is _terminal_.\n *\n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.4")\npublic inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {\n for (element in this) if
(predicate(element)) return false\n return true\n}\n\n/**\n * Returns a sequence which performs
the given [action] on each element of the original sequence as they pass through it.\n *\n * The operation is
intermediate and _stateless_.\n *\n@SinceKotlin("1.1")\npublic fun <T> Sequence<T>.onEach(action: (T) ->
Unit): Sequence<T> {\n return map {\n action(it)\n it\n }\n}\n\n/**\n * Returns a sequence which
performs the given [action] on each element of the original sequence as they pass through it.\n *\n * @param [action]
function that takes the index of an element and the element itself\n * and performs the action on the element.\n *\n *
The operation is _intermediate_ and _stateless_.\n *\n@SinceKotlin("1.4")\npublic fun <T>
Sequence<T>.onEachIndexed(action: (index: Int, T) -> Unit): Sequence<T> {\n return mapIndexed { index,
element ->\n action(index, element)\n element\n }\n}\n\n/**\n * Accumulates value starting with the first
element and applying [operation] from left to right\n * to current accumulator value and each element.\n

```

\* \n \* Throws an exception if this sequence is empty. If the sequence can be empty in an expected way, \n \* please use [reduceOrNull] instead. It returns `null` when its receiver is empty. \n \* \n \* @param [operation] function that takes current accumulator value and an element, \n \* and calculates the next accumulator value. \n \* \n \* The operation is \_terminal\_ \n \* \n \* @sample samples.collections.Collections.Aggregates.reduce \n \* \n public inline fun <S, T : S> Sequence<T>.reduce(operation: (acc: S, T) -> S): S { \n val iterator = this.iterator() \n if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.") \n var accumulator: S = iterator.next() \n while (iterator.hasNext()) { \n accumulator = operation(accumulator, iterator.next()) \n } \n return accumulator \n } \n \n /\*\* \n \* Accumulates value starting with the first element and applying [operation] from left to right \n \* to current accumulator value and each element with its index in the original sequence. \n \* \n \* Throws an exception if this sequence is empty. If the sequence can be empty in an expected way, \n \* please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty. \n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself, \n \* and calculates the next accumulator value. \n \* \n \* The operation is \_terminal\_ \n \* \n \* @sample samples.collections.Collections.Aggregates.reduce \n \* \n public inline fun <S, T : S> Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S { \n val iterator = this.iterator() \n if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.") \n var index = 1 \n var accumulator: S = iterator.next() \n while (iterator.hasNext()) { \n accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next()) \n } \n return accumulator \n } \n \n /\*\* \n \* Accumulates value starting with the first element and applying [operation] from left to right \n \* to current accumulator value and each element with its index in the original sequence. \n \* \n \* Returns `null` if the sequence is empty. \n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself, \n \* and calculates the next accumulator value. \n \* \n \* The operation is \_terminal\_ \n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull \n \* \n @SinceKotlin("1.4") \n public inline fun <S, T : S> Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? { \n val iterator = this.iterator() \n if (!iterator.hasNext()) return null \n var index = 1 \n var accumulator: S = iterator.next() \n while (iterator.hasNext()) { \n accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next()) \n } \n return accumulator \n } \n \n /\*\* \n \* Accumulates value starting with the first element and applying [operation] from left to right \n \* to current accumulator value and each element. \n \* \n \* Returns `null` if the sequence is empty. \n \* \n \* @param [operation] function that takes current accumulator value and an element, \n \* and calculates the next accumulator value. \n \* \n \* The operation is \_terminal\_ \n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull \n \* \n @SinceKotlin("1.4") \n @WasExperimental(ExperimentalStdlibApi::class) \n public inline fun <S, T : S> Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? { \n val iterator = this.iterator() \n if (!iterator.hasNext()) return null \n var accumulator: S = iterator.next() \n while (iterator.hasNext()) { \n accumulator = operation(accumulator, iterator.next()) \n } \n return accumulator \n } \n \n /\*\* \n \* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right \n \* to each element and current accumulator value that starts with [initial] value. \n \* \n \* Note that `acc` value passed to [operation] function should not be mutated; \n \* otherwise it would affect the previous value in resulting sequence. \n \* The [initial] value should also be immutable (or should not be mutated) \n \* as it may be passed to [operation] function later because of sequence's lazy nature. \n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. \n \* \n \* The operation is \_intermediate\_ and \_stateless\_ \n \* \n \* @sample samples.collections.Collections.Aggregates.runningFold \n \* \n @SinceKotlin("1.4") \n public fun <T, R> Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R): Sequence<R> { \n return sequence { \n yield(initial) \n var accumulator = initial \n for (element in this@runningFold) { \n accumulator = operation(accumulator, element) \n yield(accumulator) \n } \n } \n } \n \n /\*\* \n \* Returns a sequence containing successive

accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original sequence and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting sequence.\n \* The [initial] value should also be immutable (or should not be mutated)\n \* as it may be passed to [operation] function later because of sequence's lazy nature.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* The operation is `_intermediate_` and `_stateless_`.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningFold\n \* \n \* @SinceKotlin("1.4")\n \* \n \* public fun <T, R> Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n \* return sequence {\n \* yield(initial)\n \* var index = 0\n \* var accumulator = initial\n \* for (element in this@runningFoldIndexed) {\n \* accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n \* }\n \* }\n \* }\n \* \n \* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with the first element of this sequence.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting sequence.\n \* \n \* @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n \* \n \* The operation is `_intermediate_` and `_stateless_`.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n \* @SinceKotlin("1.4")\n \* \n \* @WasExperimental(ExperimentalStdlibApi::class)\n \* \n \* public fun <S, T : S> Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {\n \* return sequence {\n \* val iterator = iterator()\n \* if (iterator.hasNext()) {\n \* var accumulator: S = iterator.next()\n \* yield(accumulator)\n \* while (iterator.hasNext()) {\n \* accumulator = operation(accumulator, iterator.next())\n \* yield(accumulator)\n \* }\n \* }\n \* }\n \* }\n \* \n \* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original sequence and current accumulator value that starts with the first element of this sequence.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting sequence.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* The operation is `_intermediate_` and `_stateless_`.\n \* \n \* @sample samples.collections.Collections.Aggregates.runningReduce\n \* \n \* @SinceKotlin("1.4")\n \* \n \* public fun <S, T : S> Sequence<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {\n \* return sequence {\n \* val iterator = iterator()\n \* if (iterator.hasNext()) {\n \* var accumulator: S = iterator.next()\n \* yield(accumulator)\n \* var index = 1\n \* while (iterator.hasNext()) {\n \* accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n \* yield(accumulator)\n \* }\n \* }\n \* }\n \* }\n \* \n \* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n \* to each element and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting sequence.\n \* \n \* The [initial] value should also be immutable (or should not be mutated)\n \* as it may be passed to [operation] function later because of sequence's lazy nature.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n \* \n \* The operation is `_intermediate_` and `_stateless_`.\n \* \n \* @sample samples.collections.Collections.Aggregates.scan\n \* \n \* @SinceKotlin("1.4")\n \* \n \* @WasExperimental(ExperimentalStdlibApi::class)\n \* \n \* public fun <T, R> Sequence<T>.scan(initial: R, operation: (acc: R, T) -> R): Sequence<R> {\n \* return runningFold(initial, operation)\n \* }\n \* \n \* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n \* to each element, its index in the original sequence and current accumulator value that starts with [initial] value.\n \* \n \* Note that `acc` value

passed to [operation] function should not be mutated;\n \* otherwise it would affect the previous value in resulting sequence.\n \* The [initial] value should also be immutable (or should not be mutated)\n \* as it may be passed to [operation] function later because of sequence's lazy nature.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value\n \* and the element itself, and calculates the next accumulator value.\n \* \n \* The operation is \_intermediate\_ and \_stateless\_.\n \* \n \* @sample samples.collections.Collections.Aggregates.scan\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>\nSequence<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n return\n runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function\n applied to each element in the sequence.\n * \n * The operation is _terminal_.\n *\n@Deprecated("Use sumOf\n instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic\n inline fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n *\n@Deprecated("Use\n sumOf instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince =\n "1.5")\npublic inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double\n = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum\n of all values produced by [selector] function applied to each element in the sequence.\n * \n * The operation is\n _terminal_.\n
```

```
\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic\n inline fun <T> Sequence<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for\n (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n/**\n * Returns the sum of all values\n produced by [selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
```

```
\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>\nSequence<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum +=\n selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values\n produced by [selector] function\n applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
```

```
\n *\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic\n inline fun <T> Sequence<T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for\n (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n/**\n * Returns the sum of all values\n produced by [selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
```

```
\n *\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\n s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum +=\n selector(element)\n }\n return\n sum\n}\n\n/**\n * Returns the sum of\n all values produced by [selector] function applied to each element in the sequence.\n * \n * The operation is\n _terminal_.\n
```

```
\n *\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\n pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong):\n ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum +=\n selector(element)\n }\n return sum\n }\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an\n [IllegalArgumentException] if there are any `null` elements.\n * \n * The operation is _intermediate_ and\n _stateless_.\n *\npublic fun <T : Any> Sequence<T?.>.requireNonNulls(): Sequence<T> {\n return map { it ?\n
```





```

`true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n *\n * The operation is
terminal.\n *\n * @sample samples.collections.Sequences.Transformations.partition\n *\npublic inline fun <T>
Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val
second = ArrayList<T>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n
 } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Returns a sequence
containing all elements of the original sequence and then the given [element].\n *\n * The operation is
intermediate and _stateless_.\n *\npublic operator fun <T> Sequence<T>.plus(element: T): Sequence<T> {\n
return sequenceOf(this, sequenceOf(element)).flatten()\n}\n\n/**\n * Returns a sequence containing all elements of
original sequence and then all elements of the given [elements] array.\n
*\n * Note that the source sequence and the array being added are iterated only when an `iterator` is requested
from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n
*\n * The operation is _intermediate_ and _stateless_.\n *\npublic operator fun <T> Sequence<T>.plus(elements:
Array<out T>): Sequence<T> {\n return this.plus(elements.asList())\n}\n\n/**\n * Returns a sequence containing
all elements of original sequence and then all elements of the given [elements] collection.\n *\n * Note that the
source sequence and the collection being added are iterated only when an `iterator` is requested from\n * the
resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The
operation is _intermediate_ and _stateless_.\n *\npublic operator fun <T> Sequence<T>.plus(elements:
Iterable<T>): Sequence<T> {\n return sequenceOf(this, elements.asSequence()).flatten()\n}\n\n/**\n * Returns a
sequence containing all elements of original sequence and then all elements of the given [elements]
sequence.\n *\n * Note that the source sequence and the sequence being added are iterated only when an `iterator` is
requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect
the result.\n *\n * The operation is _intermediate_ and _stateless_.\n *\npublic operator fun <T>
Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {\n return sequenceOf(this,
elements).flatten()\n}\n\n/**\n * Returns a sequence containing all elements of the original sequence and then the
given [element].\n *\n * The operation is _intermediate_ and _stateless_.\n *\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.plusElement(element: T): Sequence<T> {\n return plus(element)\n}\n\n/**\n * Returns a
sequence of snapshots of the window of the given [size]\n * sliding along this sequence with the given
[step], where each\n
*\n * snapshot is a list.\n *\n * Several last lists may have fewer elements than the given [size].\n *\n * Both [size] and
[step] must be positive and can be greater than the number of elements in this sequence.\n *\n * @param size the
number of elements to take in each window\n * @param step the number of elements to move the window forward
by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the
end if any,\n * by default `false` which means partial windows won't be preserved\n *\n * @sample
samples.collections.Sequences.Transformations.takeWindows\n *\n@SinceKotlin("1.2")\npublic fun <T>
Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<List<T>> {\n return
windowedSequence(size, step, partialWindows, reuseBuffer = false)\n}\n\n/**\n * Returns a sequence of results of
applying the given [transform] function to\n * an each list representing a view over the window of the given [size]\n
*\n
*\n * sliding along this sequence with the given [step].\n *\n * Note that the list passed to the [transform] function is
ephemeral and is valid only inside that function.\n *\n * You should not store it or allow it to escape in some way,
unless you made a snapshot of it.\n *\n * Several last lists may have fewer elements than the given [size].\n *\n * Both
[size] and [step] must be positive and can be greater than the number of elements in this sequence.\n *\n * @param size
the number of elements to take in each window\n * @param step the number of elements to move the window
forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial
windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n *\n * @sample
samples.collections.Sequences.Transformations.averageWindows\n *\n@SinceKotlin("1.2")\npublic fun <T, R>
Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) ->

```

```

R): Sequence<R> {\n return windowedSequence(size, step, partialWindows, reuseBuffer =
true).map(transform)\n}\n\n/**\n * Returns a sequence of values built from the elements of `this` sequence and the
[other] sequence with the same index.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n
*\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Sequences.Transformations.zip\n */\npublic infix fun <T, R> Sequence<T>.zip(other:
Sequence<R>): Sequence<Pair<T, R>> {\n return MergingSequence(this, other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The resulting sequence ends
as soon as the shortest input sequence ends.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n *
@sample samples.collections.Sequences.Transformations.zipWithTransform\n */\npublic fun <T, R, V> Sequence<T>.zip(other: Sequence<R>, transform: (a: T, b: R) -> V): Sequence<V> {\n
return MergingSequence(this, other, transform)\n}\n\n/**\n * Returns a sequence of pairs of each two adjacent
elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two
elements.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin("1.2")\npublic fun <T>
Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {\n return zipWithNext { a, b -> a to b }\n}\n\n/**\n *
Returns a sequence containing the results of applying the given [transform] function\n * to an each pair of two
adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two
elements.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n */\n@SinceKotlin("1.2")\npublic fun <T, R> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R):
Sequence<R> {\n return sequenceResult@\n val iterator = iterator()\n if (!iterator.hasNext())
return@result\n var current = iterator.next()\n while (iterator.hasNext()) {\n val next =
iterator.next()\n yield(transform(current, next))\n current = next\n }\n }\n}\n\n/**\n * Appends
the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
*\n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first
[limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * The
operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun
<T, A : Appendable> Sequence<T>.joinTo(buffer:
A, separator: CharSequence = "\n", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1,
truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var
count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <=
limit) {\n buffer.appendElement(element, transform)\n } else break\n }\n if (limit >= 0 && count >
limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Creates a string from all
the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the
collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n
* elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * The operation is
intermediate.\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun <T> Sequence<T>.joinToString(separator: CharSequence = "\n", prefix: CharSequence = "",
postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? =
null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated,
transform).toString()\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original sequence returning its
elements when being iterated.\n */\npublic fun <T> Sequence<T>.asIterable(): Iterable<T> {\n return Iterable {
this.iterator() }\n}\n\n/**\n * Returns this sequence as a [Sequence].\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.asSequence(): Sequence<T> {\n return this\n}\n\n/**\n * Returns an average value
of elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n * @kotlin.jvm.JvmName("averageOfByte")\n */\npublic fun Sequence<Byte>.average(): Double {\n var sum:
Double = 0.0\n

```

```

 var count: Int = 0\n for (element in this) {\n sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the\n sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun\n Sequence<Short>.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum /\n count\n}\n\n/**\n * Returns an average value of elements in the\n sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Sequence<Int>.average(): Double {\n var sum: Double\n = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n checkCountOverflow(++count)\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns\n an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun Sequence<Long>.average(): Double {\n var sum:\n Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns\n an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Sequence<Float>.average(): Double {\n var sum:\n Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns\n an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Sequence<Double>.average(): Double {\n var sum:\n Double\n = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n }\n return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns\n the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Sequence<Byte>.sum(): Int {\n var sum: Int = 0\n for\n (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the\n sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun\n Sequence<Short>.sum(): Int {\n var sum: Int = 0\n for (element in this) {\n sum += element\n }\n return\n sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Sequence<Int>.sum(): Int {\n var sum: Int = 0\n for\n (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is\n _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Sequence<Long>.sum(): Long {\n var sum:\n Long = 0L\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of\n all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Sequence<Float>.sum(): Float {\n var sum: Float = 0.0f\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in\n the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun\n Sequence<Double>.sum(): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n"/**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language\n contributors.\n * Use of this\n source code is governed by the Apache 2.0 license that can be found in the\n license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n\npackage\n kotlin.collections\n\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:\n https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport\n kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns a set containing all elements of the original\n set except the given [element].\n *\n * The returned set preserves the element iteration order of the original\n set.\n */\n\npublic operator fun <T> Set<T>.minus(element: T): Set<T> {\n val result =\n LinkedHashSet<T>(mapCapacity(size))\n var removed = false\n return this.filterTo(result) { if (!removed && it

```



```

CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
 return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
@sample
samples.collections.Collections.Elements.elementAtOrNull
Returns the first character matching the given [predicate], or `null` if no such character was found.
@sample
samples.collections.Collections.Elements.find
Returns the last character matching the given [predicate], or `null` if no such character was found.
@sample
samples.collections.Collections.Elements.findLast
Returns the first character.
@throws NoSuchElementException
if the char sequence is empty.
public fun CharSequence.first(): Char {
 if (isEmpty()) throw NoSuchElementException("Char sequence is empty.")
 return this[0]
}
Returns the first character matching the given [predicate].
@throws [NoSuchElementException] if no such character is found.
public inline fun CharSequence.first(predicate: (Char) -> Boolean): Char {
 for (element in this) if (predicate(element)) return element
 throw NoSuchElementException("Char sequence contains no character matching the predicate.")
}
Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order, or throws [NoSuchElementException] if no non-null value was produced.
@sample
samples.collections.Collections.Transformations.firstNotNullOf
@SinceKotlin("1.5")
@kotlin.internal.InlineOnly
public inline fun <R : Any>
CharSequence.firstNotNullOf(transform: (Char) -> R?): R {
 return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException("No element of the char sequence was transformed to a non-null value.")
}
Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order, or `null` if no non-null value was produced.
@sample
samples.collections.Collections.Transformations.firstNotNullOfOrNull
@SinceKotlin("1.5")
@kotlin.internal.InlineOnly
public inline fun <R : Any>
CharSequence.firstNotNullOfOrNull(transform: (Char) -> R?): R? {
 for (element in this) {
 val result = transform(element)
 if (result != null) return result
 }
 return null
}
Returns the first character, or `null` if the char sequence is empty.
public fun CharSequence.firstOrNull(): Char? {
 return if (isEmpty()) null else this[0]
}
Returns the first character matching the given [predicate], or `null` if character was not found.
public inline fun CharSequence.firstOrNull(predicate: (Char) -> Boolean): Char? {
 for (element in this) if (predicate(element)) return element
 return null
}
Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
@kotlin.internal.InlineOnly
public inline fun
CharSequence.getOrNull(index: Int, defaultValue: (Int) -> Char): Char? {
 return if (index >= 0 && index <= lastIndex) get(index) else null
}
Returns index of the first character matching the given [predicate], or -1 if the char sequence does not contain such character.
public inline fun
CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {
 for (index in indices) if (predicate(this[index])) return index
 return -1
}
Returns index of the last character matching the given [predicate], or -1 if the char sequence does not contain such character.
public inline fun
CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {
 for (index in indices.reversed()) if (predicate(this[index])) return index
 return -1
}
Returns the last character.
@throws NoSuchElementException if the char sequence is empty.
@sample

```

```

samples.text.Strings.last\n * \n\npublic fun CharSequence.last(): Char {\n if (isEmpty())\n throw\n NoSuchElementException("Char sequence is empty.")\n return this[lastIndex]}\n}\n\n/**\n * Returns the last\n * character\n * matching the given [predicate].\n * \n * @throws NoSuchElementException if no such character is found.\n * \n * @sample samples.text.Strings.last\n * \n\npublic inline fun CharSequence.last(predicate: (Char) -> Boolean): Char {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return\n element\n }\n throw NoSuchElementException("Char sequence contains no character matching the\n predicate.")\n}\n\n/**\n * Returns the last character, or `null` if the char sequence is empty.\n * \n * @sample\n * samples.text.Strings.last\n * \n\npublic fun CharSequence.lastOrNull(): Char? {\n return if (isEmpty()) null else\n this[length - 1]}\n}\n\n/**\n * Returns the last character matching the given [predicate], or `null` if no such character\n * was found.\n * \n * @sample samples.text.Strings.last\n * \n\npublic inline fun CharSequence.lastOrNull(predicate:\n (Char) -> Boolean): Char? {\n for (index in this.indices.reversed()) {\n val\n element = this[index]\n if (predicate(element)) return element\n }\n return null}\n}\n\n/**\n * Returns a\n * random character from this char sequence.\n * \n * @throws NoSuchElementException if this char sequence is\n * empty.\n * \n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n\npublic inline fun CharSequence.random(): Char\n {\n return random(Random)}\n}\n\n/**\n * Returns a random character from this char sequence using the specified\n * source of randomness.\n * \n * @throws NoSuchElementException if this char sequence is empty.\n * \n\n@SinceKotlin("1.3")\n\npublic fun CharSequence.random(random: Random): Char {\n if (isEmpty())\n throw NoSuchElementException("Char sequence is empty.")\n return get(random.nextInt(length))}\n}\n\n/**\n * Returns a random character from this char sequence, or `null` if this char sequence is empty.\n * \n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n\npublic\n inline fun CharSequence.randomOrNull():\n Char? {\n return randomOrNull(Random)}\n}\n\n/**\n * Returns a random character from this char sequence\n * using the specified source of randomness, or `null` if this char sequence is empty.\n * \n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun\n CharSequence.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return\n get(random.nextInt(length))}\n}\n\n/**\n * Returns the single character, or throws an exception if the char sequence\n * is empty or has more than one character.\n * \n\npublic fun CharSequence.single(): Char {\n return when (length)\n {\n 0 -> throw NoSuchElementException("Char sequence is empty.")\n 1 -> this[0]\n else -> throw\n IllegalArgumentException("Char sequence has more than one element.")\n }\n}\n\n/**\n * Returns the single\n * character matching the given [predicate], or throws exception if there is no or more than one matching character.\n * \n\npublic inline fun CharSequence.single(predicate:\n (Char) -> Boolean): Char {\n var single: Char? = null\n var found = false\n for (element in this) {\n if\n (predicate(element)) {\n if (found) throw IllegalArgumentException("Char sequence contains more than one\n matching element.")\n single = element\n found = true\n }\n }\n if (!found) throw\n NoSuchElementException("Char sequence contains no character matching the predicate.")\n}\n\n@Suppress("UNCHECKED_CAST")\n return single as Char}\n}\n\n/**\n * Returns single character, or `null` if\n * the char sequence is empty or has more than one character.\n * \n\npublic fun CharSequence.singleOrNull(): Char?\n {\n return if (length == 1) this[0] else null}\n}\n\n/**\n * Returns the single character matching the given\n * [predicate], or `null` if character was not found or more than one character was found.\n * \n\npublic inline fun\n CharSequence.singleOrNull(predicate: (Char) -> Boolean): Char? {\n var single:\n Char? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found)\n return null\n single = element\n found = true\n }\n }\n if (!found) return null\n return\n single}\n}\n\n/**\n * Returns a subsequence of this char sequence with the first [n] characters removed.\n * \n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n * \n\npublic fun\n CharSequence.drop(n: Int): CharSequence {\n require(n >= 0) { "Requested character count $n is less than zero." }\n return subSequence(n.coerceAtMost(length), length)}\n}\n\n/**\n * Returns a string with the first [n] characters\n * removed.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample

```

```

samples.text.Strings.drop\n *\npublic fun String.drop(n: Int): String {\n require(n >= 0) { \"Requested character
count $n is less than zero.\" }\n return substring(n.coerceAtMost(length))\n}\n\n/**\n
* Returns a subsequence of this char sequence with the last [n] characters removed.\n * \n * @throws
IllegalArgumentOutOfRangeException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n *\npublic fun
CharSequence.dropLast(n: Int): CharSequence {\n require(n >= 0) { \"Requested character count $n is less than
zero.\" }\n return take((length - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a string with the last [n] characters
removed.\n * \n * @throws IllegalArgumentOutOfRangeException if [n] is negative.\n * \n * @sample
samples.text.Strings.drop\n *\npublic fun String.dropLast(n: Int): String {\n require(n >= 0) { \"Requested
character count $n is less than zero.\" }\n return take((length - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a
subsequence of this char sequence containing all characters except last characters that satisfy the given [predicate].\n
*\n * @sample samples.text.Strings.drop\n *\npublic inline fun CharSequence.dropLastWhile(predicate: (Char) ->
Boolean):
CharSequence {\n for (index in lastIndex downTo 0)\n if (!predicate(this[index]))\n return
subSequence(0, index + 1)\n return \"\"\n}\n\n/**\n * Returns a string containing all characters except last
characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n *\npublic inline fun
String.dropLastWhile(predicate: (Char) -> Boolean): String {\n for (index in lastIndex downTo 0)\n if
(!predicate(this[index]))\n return substring(0, index + 1)\n return \"\"\n}\n\n/**\n * Returns a subsequence
of this char sequence containing all characters except first characters that satisfy the given [predicate].\n * \n *
@sample samples.text.Strings.drop\n *\npublic inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean):
CharSequence {\n for (index in this.indices)\n if (!predicate(this[index]))\n return subSequence(index,
length)\n return \"\"\n}\n\n/**\n * Returns a string containing
all characters except first characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n
*\npublic inline fun String.dropWhile(predicate: (Char) -> Boolean): String {\n for (index in this.indices)\n if
(!predicate(this[index]))\n return substring(index)\n return \"\"\n}\n\n/**\n * Returns a char sequence
containing only those characters from the original char sequence that match the given [predicate].\n * \n * @sample
samples.text.Strings.filter\n *\npublic inline fun CharSequence.filter(predicate: (Char) -> Boolean): CharSequence
{\n return filterTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from
the original string that match the given [predicate].\n * \n * @sample samples.text.Strings.filter\n *\npublic inline
fun String.filter(predicate: (Char) -> Boolean): String {\n return filterTo(StringBuilder(),
predicate).toString()\n}\n\n/**\n * Returns a char sequence containing
only those characters from the original char sequence that match the given [predicate].\n * @param [predicate]
function that takes the index of a character and the character itself\n * and returns the result of predicate evaluation
on the character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun
CharSequence.filterIndexed(predicate: (index: Int, Char) -> Boolean): CharSequence {\n return
filterIndexedTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from the
original string that match the given [predicate].\n * @param [predicate] function that takes the index of a character
and the character itself\n * and returns the result of predicate evaluation on the character.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n *\npublic inline fun String.filterIndexed(predicate: (index:
Int, Char) -> Boolean): String {\n return filterIndexedTo(StringBuilder(), predicate).toString()\n}\n\n/**\n
* Appends all characters matching the given [predicate] to the given [destination].\n * @param [predicate] function
that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the
character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n *\npublic inline fun <C :
Appendable> CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n
 forEachIndexed { index, element ->\n if (predicate(index, element)) destination.append(element)\n }\n return destination\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char
sequence that do not match the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n *\npublic inline
fun CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {\n return filterNotTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only

```

```

those characters from the original string that do not match the given [predicate].\n * \n * @sample
samples.text.Strings.filterNot\n * \n\npublic inline fun String.filterNot(predicate: (Char) -> Boolean): String {\n
return filterNotTo(StringBuilder(), predicate).toString()\n}\n\n/**\n * Appends all characters not matching the given
[predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n\npublic inline fun <C : Appendable> CharSequence.filterNotTo(destination: C, predicate: (Char) -> Boolean): C
{\n for (element in this) if (!predicate(element)) destination.append(element)\n return destination\n}\n\n/**\n *
Appends all characters matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n * \n\npublic inline fun <C : Appendable>
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {\n for (index in 0 until length) {\n val
element
= get(index)\n if (predicate(element)) destination.append(element)\n }\n return destination\n}\n\n/**\n *
Returns a char sequence containing characters of the original char sequence at the specified range of [indices].\n
*\n\npublic fun CharSequence.slice(indices: IntRange): CharSequence {\n if (indices.isEmpty()) return ""\n
return subSequence(indices)\n}\n\n/**\n * Returns a string containing characters of the original string at the
specified range of [indices].\n * \n\npublic fun String.slice(indices: IntRange): String {\n if (indices.isEmpty())
return ""\n return substring(indices)\n}\n\n/**\n * Returns a char sequence containing characters of the original
char sequence at specified [indices].\n * \n\npublic fun CharSequence.slice(indices: Iterable<Int>): CharSequence {\n
val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return ""\n val result = StringBuilder(size)\n
for (i in indices) {\n result.append(get(i))\n }\n return result\n}\n\n/**\n * Returns a string containing
characters of the original string at specified [indices].\n
*\n\n@kotlin.internal.InlineOnly\n\npublic inline fun String.slice(indices: Iterable<Int>): String {\n return (this as
CharSequence).slice(indices).toString()\n}\n\n/**\n * Returns a subsequence of this char sequence containing the
first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n * \n\npublic fun
CharSequence.take(n: Int): CharSequence {\n require(n >= 0) { "Requested character count $n is less than zero." }\n
return subSequence(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a string containing the first [n]
characters from this string, or the entire string if this string is shorter.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.text.Strings.take\n * \n\npublic
fun String.take(n: Int): String {\n require(n >= 0) { "Requested character count $n is less than zero." }\n return
substring(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing the last
[n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n * \n\npublic fun
CharSequence.takeLast(n: Int): CharSequence {\n require(n >= 0) { "Requested character count $n is less than
zero." }\n val length = length\n return subSequence(length - n.coerceAtMost(length), length)\n}\n\n/**\n *
Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.\n
*\n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n * \n\npublic fun
String.takeLast(n: Int): String {\n require(n >= 0) { "Requested character
count $n is less than zero." }\n val length = length\n return substring(length -
n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing last characters that
satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n * \n\npublic inline fun
CharSequence.takeLastWhile(predicate: (Char) -> Boolean): CharSequence {\n for (index in lastIndex downTo 0)
{\n if (!predicate(this[index])) {\n return subSequence(index + 1, length)\n }\n }\n return
subSequence(0, length)\n}\n\n/**\n * Returns a string containing last characters that satisfy the given [predicate].\n
*\n * @sample samples.text.Strings.take\n * \n\npublic inline fun String.takeLastWhile(predicate: (Char) ->
Boolean): String {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return
substring(index + 1)\n }\n }\n return this\n}\n\n/**\n * Returns a subsequence of this char sequence
containing the first characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n
*\n\npublic inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean): CharSequence {\n for (index in 0

```



```

until length)\n if (!predicate(get(index))) {\n return subSequence(0, index)\n }\n return
subSequence(0, length)\n}\n\n/**\n * Returns a string containing the first characters that satisfy the given
[predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic inline fun String.takeWhile(predicate: (Char) ->
Boolean): String {\n for (index in 0 until length)\n if (!predicate(get(index))) {\n return substring(0,
index)\n }\n return this\n}\n\n/**\n * Returns a char sequence with characters in reversed order.\n */\npublic
fun CharSequence.reversed(): CharSequence {\n return StringBuilder(this).reverse()\n}\n\n/**\n * Returns a string
with characters in reversed order.\n */\n@kotlin.internal.InlineOnly\npublic
inline fun String.reversed(): String {\n return (this as CharSequence).reversed().toString()\n}\n\n/**\n * Returns a
[Map] containing key-value pairs provided by [transform] function\n * applied to characters of the given char
sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associate\n */\npublic inline fun <K, V> CharSequence.associate(transform: (Char) -> Pair<K,
V>): Map<K, V> {\n val capacity = mapCapacity(length).coerceAtLeast(16)\n return
associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the characters
from the given char sequence indexed by the key\n * returned from [keySelector] function applied to each
character.\n * \n * If any two characters would have the same key returned by [keySelector] the last one gets added
to the map.\n *
\n * The returned map preserves the entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associateBy\n */\npublic inline fun <K> CharSequence.associateBy(keySelector: (Char) -> K):
Map<K, Char> {\n val capacity = mapCapacity(length).coerceAtLeast(16)\n return
associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
values provided by [valueTransform] and indexed by [keySelector] functions applied to characters of the given char
sequence.\n * \n * If any two characters would have the same key returned by [keySelector] the last one gets added
to the map.\n * \n * The returned map preserves the entry iteration order of the original char sequence.\n * \n *
@sample samples.text.Strings.associateByWithValueTransform\n */\npublic inline fun <K, V>
CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n val capacity
= mapCapacity(length).coerceAtLeast(16)\n
 return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
function applied to each character of the given char sequence\n * and value is the character itself.\n * \n * If any two
characters would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.text.Strings.associateByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Char>>
CharSequence.associateByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in this) {\n
destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and
value is provided by the [valueTransform] function applied to characters of the given char sequence.\n * \n *
\n * If any two characters would have the same key returned by [keySelector] the last one gets added to the map.\n *
\n * @sample samples.text.Strings.associateByToWithValueTransform\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n for (element in this) {\n destination.put(keySelector(element),
valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs\n * provided by [transform] function applied to each character of the given char
sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.text.Strings.associateTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>>
CharSequence.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {\n for (element in this) {\n
destination += transform(element)\n
 }\n return destination\n}\n\n/**\n * Returns a [Map] where keys are characters from the given char sequence
and values are\n * produced by the [valueSelector] function applied to each character.\n * \n * If any two characters

```

are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original char sequence.

```

@sample samples.text.Strings.associateWith
^@SinceKotlin("1.3")
public inline fun <V> CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {
 val result = LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16))
 return result.associateWithTo(this, valueSelector)
}

```

Populates and returns the [destination] mutable map with key-value pairs for each character of the given char sequence, where key is the character itself and value is provided by the [valueSelector] function applied to that key. If any two characters are equal, the last one overwrites the former value in the map.

```

@sample samples.text.Strings.associateWithTo
^@SinceKotlin("1.3")
public inline fun <V, M : MutableMap<in Char, in V>> CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
 for (element in this) {
 destination.put(element, valueSelector(element))
 }
 return destination
}

```

Appends all characters to the given [destination] collection.

```

public fun <C : MutableCollection<in Char>> CharSequence.addToCollection(destination: C): C {
 for (item in this) {
 destination.add(item)
 }
 return destination
}

```

Returns a new [HashSet] of all characters.

```

public fun CharSequence.toHashSet(): HashSet<Char> {
 return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))
}

```

Returns a [List] containing all characters.

```

public fun CharSequence.toList(): List<Char> {
 return when (length) {
 0 -> emptyList()
 1 -> listOf(this[0])
 else -> this.toMutableList()
 }
}

```

Returns a new [MutableList] filled with all characters of this char sequence.

```

public fun CharSequence.toMutableList(): MutableList<Char> {
 return toCollection(ArrayList<Char>(length))
}

```

Returns a [Set] of all characters. The returned set preserves the element iteration order of the original char sequence.

```

public fun CharSequence.toSet(): Set<Char> {
 return when (length) {
 0 -> emptySet()
 1 -> setOf(this[0])
 else -> toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))
 }
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each character of original char sequence.

```

@sample samples.collections.Collections.Transformations.flatMap
^@public inline fun <R> CharSequence.flatMap(transform: (Char) -> Iterable<R>): List<R> {
 return flatMapTo(ArrayList<R>(), transform)
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each character and its index in the original char sequence.

```

@sample samples.collections.Collections.Transformations.flatMapIndexed
^@SinceKotlin("1.4")
^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
^@OverloadResolutionByLambdaReturnType
^@kotlin.jvm.JvmName("flatMapIndexedIterable")
^@kotlin.internal.InlineOnly
public inline fun <R> CharSequence.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {
 return flatMapIndexedTo(ArrayList<R>(), transform)
}

```

Appends all elements yielded from results of [transform] function being invoked on each character and its index in the original char sequence, to the given [destination].

```

^@SinceKotlin("1.4")
^@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
^@OverloadResolutionByLambdaReturnType
^@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
^@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {
 var index = 0
 for (element in this) {
 val list = transform(index++, element)
 destination.addAll(list)
 }
 return destination
}

```

Appends all elements yielded from results of [transform] function being invoked on each character of original char sequence, to the given [destination].

```

^@public inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
 for (element in this) {
 val list = transform(element)
 destination.addAll(list)
 }
 return destination
}

```

Groups characters of the original char sequence by the key returned by the given [keySelector] function applied to each character and returns a map where each group key is associated with

a list of corresponding characters.

```

 * The returned map preserves the entry iteration order of the keys produced
 * from the original char sequence.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
 public inline fun <K> CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {
 return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
 }
 */
 * Groups values returned by the [valueTransform] function applied to each character of the original char sequence
 * by the key returned by the given [keySelector] function applied to the character
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced
 * from the original char sequence.
 * @sample
 samples.collections.Collections.Transformations.groupByKeysAndValues
 */
 public inline fun <K, V> CharSequence.groupBy(keySelector: (Char) -> K, valueTransform: (Char)
 -> V): Map<K, List<V>> {
 return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
 }
 */
 * Groups characters of the original char sequence by the key returned by the given
 [keySelector] function
 * applied to each character and puts to the [destination] map each group key associated
 with a list of corresponding characters.
 * @return The [destination] map.
 * @sample
 samples.collections.Collections.Transformations.groupBy
 */
 public inline fun <K, M : MutableMap<in K, MutableList<Char>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M {
 for (element in this) {
 val key = keySelector(element)
 val list = destination.getOrPut(key) { ArrayList<Char>() }
 list.add(element)
 }
 return destination
 }
 */
 * Groups values returned by the [valueTransform] function applied to each character of the original char sequence
 * by the key returned by the given [keySelector] function applied
 to the character
 * and puts to the [destination] map each group key associated with a list of corresponding
 values.
 * @return The [destination] map.
 * @sample
 samples.collections.Collections.Transformations.groupByKeysAndValues
 */
 public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
 for (element in this) {
 val key = keySelector(element)
 val list = destination.getOrPut(key) { ArrayList<V>() }
 list.add(valueTransform(element))
 }
 return destination
 }
 */
 * Creates a [Grouping] source from a char sequence to be used later with one of group-and-fold operations
 * using the specified [keySelector] function to extract a key from each character.
 * @sample
 samples.collections.Grouping.groupingByEachCount
 */
 @SinceKotlin("1.1")
 public inline fun <K> CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> {
 return object : Grouping<Char, K> {
 override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()
 override fun keyOf(element: Char): K = keySelector(element)
 }
 }
 */
 * Returns a list containing the results of applying the given [transform] function
 * to each character in the original char sequence.
 * @sample samples.text.Strings.map
 */
 public inline fun <R> CharSequence.map(transform: (Char) -> R): List<R> {
 return mapTo(ArrayList<R>(length), transform)
 }
 */
 * Returns a list containing the results of applying the given [transform] function
 * to each character and its index in the original char sequence.
 * @param [transform] function that takes the index of a character and the character itself
 * and returns the result of the transform applied to the character.
 */
 public inline fun <R> CharSequence.mapIndexed(transform: (index: Int, Char) -> R): List<R> {
 return mapIndexedTo(ArrayList<R>(length), transform)
 }
 */
 * Returns a list containing only the non-null results of applying the given [transform]
 function
 * to each character and its index in the original char sequence.
 * @param [transform] function that takes the index of a character and the character itself
 * and returns the result of the transform applied to the character.
 */
 public inline fun <R : Any> CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {
 return mapIndexedNotNullTo(ArrayList<R>(), transform)
 }
 */
 * Applies the given [transform] function to each character and its index in the original char sequence
 * and appends only the non-null results to the given [destination].
 * @param [transform] function that takes the index of a character and the character itself
 * and returns the result of the transform applied to the character.
 */
 public inline fun <R : Any, D> CharSequence.mapIndexedNotNullTo(destination: D, transform: (index: Int, Char) -> R?): D {
 return destination.addAll(mapIndexedNotNullTo(ArrayList<R>(), transform))
 }
 */

```

```

C : MutableCollection<in R>> CharSequence.mapIndexedNotNullTo(destination:
C, transform: (index: Int, Char) -> R?): C { \n forEachIndexed { index, element -> transform(index, element)?.let
{ destination.add(it) } } \n return destination \n } \n \n /** \n * Applies the given [transform] function to each character
and its index in the original char sequence \n * and appends the results to the given [destination]. \n * @param
[transform] function that takes the index of a character and the character itself \n * and returns the result of the
transform applied to the character. \n */ \n public inline fun <R, C : MutableCollection<in R>>
CharSequence.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C { \n var index = 0 \n for (item
in this) \n destination.add(transform(index++, item)) \n return destination \n } \n \n /** \n * Returns a list
containing only the non-null results of applying the given [transform] function \n * to each character in the original
char sequence. \n * \n * @sample samples.collections.Collections.Transformations.mapNotNull \n
*/ \n public inline fun <R : Any> CharSequence.mapNotNull(transform: (Char) -> R?): List<R> { \n return
mapNotNullTo(ArrayList<R>(), transform) \n } \n \n /** \n * Applies the given [transform] function to each character
in the original char sequence \n * and appends only the non-null results to the given [destination]. \n */ \n public inline
fun <R : Any, C : MutableCollection<in R>> CharSequence.mapNotNullTo(destination: C, transform: (Char) ->
R?): C { \n forEach { element -> transform(element)?.let { destination.add(it) } } \n return destination \n } \n \n /** \n
* Applies the given [transform] function to each character of the original char sequence \n * and appends the results
to the given [destination]. \n */ \n public inline fun <R, C : MutableCollection<in R>>
CharSequence.mapTo(destination: C, transform: (Char) -> R): C { \n for (item in this) \n
destination.add(transform(item)) \n return destination \n } \n \n /** \n * Returns a lazy [Iterable] that wraps each
character of the original
char sequence \n * into an [IndexedValue] containing the index of that character and the character itself. \n */ \n public
fun CharSequence.withIndex(): Iterable<IndexedValue<Char>> { \n return IndexingIterable { iterator()
} \n } \n \n /** \n * Returns `true` if all characters match the given [predicate]. \n * \n * Note that if the char sequence
contains no characters, the function returns `true` \n * because there are no characters in it that do not match the
predicate. \n * See a more detailed explanation of this logic concept in ["Vacuous
truth"](https://en.wikipedia.org/wiki/Vacuous_truth) article. \n * \n * @sample
samples.collections.Collections.Aggregates.all \n */ \n public inline fun CharSequence.all(predicate: (Char) ->
Boolean): Boolean { \n for (element in this) if (!predicate(element)) return false \n return true \n } \n \n /** \n
* Returns `true` if char sequence has at least one character. \n * \n * @sample
samples.collections.Collections.Aggregates.any \n */ \n public fun CharSequence.any():
Boolean { \n return !isEmpty() \n } \n \n /** \n * Returns `true` if at least one character matches the given
[predicate]. \n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate \n */ \n public inline fun
CharSequence.any(predicate: (Char) -> Boolean): Boolean { \n for (element in this) if (predicate(element)) return
true \n return false \n } \n \n /** \n * Returns the length of this char sequence. \n
*/ \n @kotlin.internal.InlineOnly \n public inline fun CharSequence.count(): Int { \n return length \n } \n \n /** \n
* Returns the number of characters matching the given [predicate]. \n */ \n public inline fun
CharSequence.count(predicate: (Char) -> Boolean): Int { \n var count = 0 \n for (element in this) if
(predicate(element)) ++count \n return count \n } \n \n /** \n * Accumulates value starting with [initial] value and
applying [operation] from left to right \n * to current accumulator value and each character. \n * \n * Returns the
specified [initial] value if the char sequence
is empty. \n * \n * @param [operation] function that takes current accumulator value and a character, and calculates
the next accumulator value. \n */ \n public inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) ->
R): R { \n var accumulator = initial \n for (element in this) accumulator = operation(accumulator, element) \n
return accumulator \n } \n \n /** \n * Accumulates value starting with [initial] value and applying [operation] from left
to right \n * to current accumulator value and each character with its index in the original char sequence. \n * \n *
Returns the specified [initial] value if the char sequence is empty. \n * \n * @param [operation] function that takes
the index of a character, current accumulator value \n * and the character itself, and calculates the next accumulator
value. \n */ \n public inline fun <R> CharSequence.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R):

```

```

R {
 var index = 0
 var accumulator = initial
 for (element
in this) accumulator = operation(index++, accumulator, element)
 return accumulator
}

/** Accumulates
value starting with [initial] value and applying [operation] from right to left
* to each character and current
accumulator value.
* Returns the specified [initial] value if the char sequence is empty.
* @param
[operation] function that takes a character and current accumulator value, and calculates the next accumulator
value.
*/
public inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
 var index = lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(get(index--),
accumulator)
 }
 return accumulator
}

/** Accumulates value starting with [initial] value and applying
[operation] from right to left
* to each character with its index in the original char sequence and current
accumulator value.
* Returns the specified [initial] value if the
char sequence is empty.
* @param [operation] function that takes the index of a character, the character
itself
* and current accumulator value, and calculates the next accumulator value.
*/
public inline fun <R>
CharSequence.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
 var index =
lastIndex
 var accumulator = initial
 while (index >= 0) {
 accumulator = operation(index, get(index),
accumulator)
 --index
 }
 return accumulator
}

/** Performs the given [action] on each
character.
*/
public inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {
 for (element in this)
action(element)
}

/** Performs the given [action] on each character, providing sequential index with the
character.
* @param [action] function that takes the index of a character and the character itself
* and performs
the action on the character.
*/
public inline fun CharSequence.forEachIndexed(action: (index:
Int, Char) -> Unit): Unit {
 var index = 0
 for (item in this) action(index++, item)
}

/** Returns the
largest character.
* @throws NoSuchElementException if the char sequence is empty.
*/
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("maxOrThrow")
@Suppress("CONFLICTING_OVERLOADS")
public fun CharSequence.max(): Char {
 if (isEmpty()) throw NoSuchElementException()
 var max =
this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (max < e) max = e
 }
 return max
}

/** Returns the first character yielding the largest value of the given function.
* @throws
NoSuchElementException if the char sequence is empty.
* @sample
samples.collections.Collections.Aggregates.maxBy
*/
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("maxByOrThrow")
@Suppress("CONFLICTING_OVERLOADS")
public inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char {
 if
(isEmpty()) throw NoSuchElementException()

 var maxElem = this[0]
 val lastIndex = this.lastIndex
 if (lastIndex == 0) return maxElem
 var maxValue =
selector(maxElem)
 for (i in 1..lastIndex) {
 val e = this[i]
 val v = selector(e)
 if (maxValue <
v) {
 maxElem = e
 maxValue = v
 }
 }
 return maxElem
}

/** Returns the first
character yielding the largest value of the given function or `null` if there are no characters.
* @sample
samples.collections.Collections.Aggregates.maxByOrNull
*/
@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> CharSequence.maxByOrNull(selector: (Char) -> R): Char? {
 if (isEmpty()) return null
 var
maxElem = this[0]
 val lastIndex = this.lastIndex
 if (lastIndex == 0) return maxElem
 var maxValue =
selector(maxElem)
 for (i in 1..lastIndex) {
 val e = this[i]
 val v = selector(e)
 if (maxValue <
v) {
 maxElem = e
 maxValue = v
 }
 }
 return maxElem
}

/** Returns the largest value among all values
produced by [selector] function
* applied to each character in the char sequence.
* @sample
samples.collections.Collections.Aggregates.maxOf
* If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.
* @throws NoSuchElementException if the char
sequence is empty.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharSequence.maxOf(selector: (Char) ->
Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var maxValue = selector(this[0])
 for
(i in 1..lastIndex) {
 val v = selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return
maxValue
}

/** Returns the largest value among all values produced by [selector] function
* applied to
*/

```

each character in the char sequence.  
 \* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.  
 \* @throws NoSuchElementException if the char sequence is empty.

```

*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun CharSequence.maxOf(selector: (Char) ->
Float): Float {\/n if (isEmpty()) throw NoSuchElementException()\/n var maxVAlue = selector(this[0])\/n for (i
in 1..lastIndex) {\/n val v = selector(this[i])\/n maxVAlue = maxOf(maxVAlue, v)\/n }\/n return
maxVAlue\/n}\/n\/n**\/n * Returns the largest value among all values produced by [selector] function\/n * applied to
each character in the char sequence.\/n *\/n * @throws NoSuchElementException if the char sequence is empty.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun <R : Comparable<R>>
CharSequence.maxOf(selector:
(Char) -> R): R {\/n if (isEmpty()) throw NoSuchElementException()\/n var maxVAlue = selector(this[0])\/n for
(i in 1..lastIndex) {\/n val v = selector(this[i])\/n if (maxVAlue < v) {\/n maxVAlue = v\/n }\/n }\/n
return maxVAlue\/n}\/n\/n**\/n * Returns the largest value among all values produced by [selector] function\/n *
applied to each character in the char sequence or `null` if there are no characters.\/n *\/n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Double): Double? {\/n if (isEmpty()) return null\/n var maxVAlue = selector(this[0])\/n for (i in
1..lastIndex) {\/n val v = selector(this[i])\/n maxVAlue = maxOf(maxVAlue, v)\/n }\/n return
maxVAlue\/n}\/n\/n**\/n * Returns the largest value among all values produced by [selector] function\/n * applied to
each character in the char sequence or `null` if there are no characters.\/n *\/n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Float): Float? {\/n if (isEmpty()) return null\/n var maxVAlue = selector(this[0])\/n for (i in
1..lastIndex) {\/n val v = selector(this[i])\/n maxVAlue = maxOf(maxVAlue, v)\/n }\/n return
maxVAlue\/n}\/n\/n**\/n * Returns the largest value among all values produced by [selector] function\/n * applied to
each character in the char sequence or `null` if there are no characters.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic
inline fun <R : Comparable<R>> CharSequence.maxOfOrNull(selector: (Char) -> R): R? {\/n if (isEmpty()) return
null\/n var maxVAlue = selector(this[0])\/n for (i in 1..lastIndex) {\/n val v = selector(this[i])\/n if
(maxVAlue < v) {\/n maxVAlue = v\/n }\/n }\/n return maxVAlue\/n}\/n\/n**\/n * Returns the largest value
according to the provided [comparator]\/n * among all values produced by [selector] function applied to each
character in the char sequence.\/n *\/n * @throws NoSuchElementException if the char sequence is empty.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun <R> CharSequence.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\/n if (isEmpty()) throw NoSuchElementException()\/n var
maxVAlue = selector(this[0])\/n for
(i in 1..lastIndex) {\/n val v = selector(this[i])\/n if (comparator.compare(maxVAlue, v) < 0) {\/n
maxVAlue = v\/n }\/n }\/n return maxVAlue\/n}\/n\/n**\/n * Returns the largest value according to the provided
[comparator]\/n * among all values produced by [selector] function applied to each character in the char sequence or
`null` if there are no characters.\/n
*\/n@SinceKotlin("1.4")\/n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/n@OverloadResolution
ByLambdaReturnType\/n@kotlin.internal.InlineOnly\/npublic inline fun <R>
CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\/n if (isEmpty())

```

```

return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\nReturns the largest character or `null` if there are no characters.\n*/\n\n@SinceKotlin("1.4")\npublic fun CharSequence.maxOrNull(): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\nReturns the first character having the largest value according to the provided [comparator].\n*/\n\n@throws NoSuchElementException if the char sequence is empty.\n\n*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharSequence.maxWith(comparator: Comparator<in Char>): Char {\n if (isEmpty())\n throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\nReturns the first character having the largest value according to the provided [comparator] or `null` if there are no characters.\n\n*/\n\n@SinceKotlin("1.4")\npublic fun CharSequence.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\nReturns the smallest character.\n*/\n\n@throws NoSuchElementException if the char sequence is empty.\n\n*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharSequence.min(): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\nReturns the first character yielding the smallest value of the given function.\n*/\n\n@throws NoSuchElementException if the char sequence is empty.\n*/\n\n@sample samples.collections.Collections.Aggregates.minBy\n\n*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> CharSequence.minBy(selector: (Char) -> R): Char {\n if (isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\nReturns the first character yielding the smallest value of the given function or `null` if there are no characters.\n*/\n\n@sample samples.collections.Collections.Aggregates.minByOrNull\n\n*/\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\nReturns the smallest value among all values produced by [selector] function\n*/\n\napplied to each character in the char sequence.\n*/\n\nIf any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n*/\n\n@throws NoSuchElementException if the char sequence is empty.\n\n*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\nReturns the smallest value among all values produced by [selector] function\n*/\n\napplied to each character in the char sequence.\n*/\n\nIf any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n*/\n\n@throws NoSuchElementException if the char sequence is empty.\n\n*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->

```

```

Float): Float {
 if (isEmpty()) throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return
 minValue
}

Returns the smallest value among all values produced by [selector] function
* applied to each character in the char sequence.
* @throws NoSuchElementException if the char sequence is empty.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun <R : Comparable<R>> CharSequence.minOf(selector: (Char) -> R): R {
 if (isEmpty()) throw
 NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v =
 selector(this[i])
 if (minValue > v) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value among all values produced by [selector] function
* applied to each character in the char
sequence or `null` if there are no characters.
* If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun CharSequence.minOfOrNull(selector:
(Char) -> Double): Double? {
 if (isEmpty()) return null
 var
 minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 minValue =
 minOf(minValue, v)
 }
 return minValue
}

Returns the smallest value among all values produced
by [selector] function
* applied to each character in the char sequence or `null`
if there are no characters.
* If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun CharSequence.minOfOrNull(selector:
(Char) -> Float): Float? {
 if (isEmpty()) return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex)
 {
 val v = selector(this[i])
 minValue = minOf(minValue, v)
 }
 return minValue
}

Returns the smallest value among all values produced by [selector] function
* applied to each character
in the char sequence or `null` if there are no characters.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun <R : Comparable<R>>
CharSequence.minOfOrNull(selector: (Char) -> R): R? {
 if (isEmpty()) return null
 var minValue =
 selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (minValue > v) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided
[comparator]
* among all values produced by [selector] function applied to each character in the char sequence.
* @throws NoSuchElementException if the char sequence is empty.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun <R> CharSequence.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {
 if (isEmpty()) throw NoSuchElementException()
 var
 minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if
 (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator]
* among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.

SinceKotlin("1.4")
OptIn(kotlin.experimental.ExperimentalTypeInference::class)
OverloadResolution
ByLambdaReturnType
kotlin.internal.InlineOnly
public
inline fun <R>
CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
 if (isEmpty())
 return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if
 (comparator.compare(minValue, v) > 0)
 {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest character or `null` if
there are no characters.

SinceKotlin("1.4")
public fun CharSequence.minOrNull(): Char? {
 if
 (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (min > e) min =
 e
 }
 return min
}

Returns the first character having the smallest value according to the provided

```



```

[comparator].\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharSequence.minWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw\n NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if\n (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first character having\n the smallest value according to the provided [comparator] or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\npublic fun CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char?\n {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if\n (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if the char sequence has\n no characters.\n
*\n * @sample samples.collections.Collections.Aggregates.none\n
*\npublic fun\n CharSequence.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if no characters match the given\n [predicate].\n
*\n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\npublic inline fun\n CharSequence.none(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return\n false\n return true\n}\n\n/**\n * Performs the given [action] on each character and returns the char sequence\n itself afterwards.\n
*\n@SinceKotlin("1.1")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) ->\n Unit): S {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on\n each character, providing sequential index with the character,\n * and returns the char sequence itself afterwards.\n
*\n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on\n the character.\n
*\n@SinceKotlin("1.4")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index:\n Int, Char) -> Unit): S {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with\n the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n
*\n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver\n is empty.\n
*\n * @param [operation] function that takes current accumulator value and a character,\n * and\n calculates the next accumulator value.\n
*\n * @sample samples.collections.Collections.Aggregates.reduce\n
*\npublic inline fun CharSequence.reduce(operation: (acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return\n accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to\n right\n * to current accumulator value and each character with its index in the original char sequence.\n
*\n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n
*\n * @param [operation] function that takes the index of a character, current accumulator value and the character\n itself,\n * and calculates the next accumulator value.\n
*\n * @sample\n samples.collections.Collections.Aggregates.reduce\n
*\npublic inline fun CharSequence.reduceIndexed(operation:\n (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw\n UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = this[0]\n for\n (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return\n accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to\n right\n * to current accumulator value and each character with its index in the original char sequence.\n
*\n * Returns `null` if the char sequence is empty.\n
*\n * @param [operation] function that takes the index of a\n character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n
*\n * @sample\n samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun\n CharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,\n accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first\n character and applying [operation] from left to right\n * to current accumulator value and each character.\n
*\n *

```

Returns `null` if the char sequence is empty.

`\n * \n * @param [operation] function that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the last character and applying [operation] from right to left\n * \n * to each character and current accumulator value.\n * \n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * \n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * \n * @param [operation] function that takes a character and current accumulator value,\n * \n * and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * \n * public inline fun CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n * \n * var index = lastIndex\n * \n * if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n * \n * var accumulator = get(index--)\n * \n * while (index >= 0) {\n * \n * accumulator = operation(get(index--), accumulator)\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the last character and applying [operation] from right to left\n * \n * to each character with its index in the original char sequence and current accumulator value.\n * \n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * \n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * \n * @param [operation] function that takes the index of a character, the character itself and current accumulator value,\n * \n * and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * \n * public inline fun CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n * \n * var index = lastIndex\n * \n * if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n * \n * var accumulator = get(index--)\n * \n * while (index >= 0) {\n * \n * accumulator = operation(index, get(index), accumulator)\n * \n * --index\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the last character and applying [operation] from right to left\n * \n * to each character with its index in the original char sequence and current accumulator value.\n * \n * \n * Returns `null` if the char sequence is empty.\n * \n * \n * @param [operation] function that takes the index of a character, the character itself and current accumulator value,\n * \n * and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * \n * public inline fun CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n * \n * var index = lastIndex\n * \n * if (index < 0) return null\n * \n * var accumulator = get(index--)\n * \n * while (index >= 0) {\n * \n * accumulator = operation(index, get(index), accumulator)\n * \n * --index\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the last character and applying [operation] from right to left\n * \n * to each character and current accumulator value.\n * \n * \n * Returns `null` if the char sequence is empty.\n * \n * \n * @param [operation] function that takes a character and current accumulator value,\n * \n * and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * \n * public inline fun CharSequence.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n * \n * var index = lastIndex\n * \n * if (index < 0) return null\n * \n * var accumulator = get(index--)\n * \n * while (index >= 0) {\n * \n * accumulator = operation(get(index--), accumulator)\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * \n * to each character and current accumulator value that starts with [initial] value.\n * \n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * \n * otherwise it would affect the previous value in resulting list.\n * \n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n * \n * \n * public inline fun <R> CharSequence.runningFold(initial: R, operation: (acc: R, Char) ->`





characters than the given [size].\n \* \n \* @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @sample samples.text.Strings.chunkedTransform\n \* \n \* @SinceKotlin("1.2")\n public fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {\n return windowed(size, size, partialWindows = true, transform = transform)\n }\n\n\n \* Splits this char sequence into a sequence of strings each not exceeding the given [size].\n \* \n \* The last string in the resulting sequence may have fewer characters than the given [size].\n \* \n \* @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @sample samples.collections.Collections.Transformations.chunked\n \* \n \* @SinceKotlin("1.2")\n public fun CharSequence.chunkedSequence(size: Int): Sequence<String> {\n return chunkedSequence(size) { it.toString() }\n }\n\n\n \* Splits this char sequence into several char sequences each not exceeding the given [size]\n \* and applies the given [transform] function to an each.\n \* \n \* @return sequence of results of the [transform] applied to an each char sequence.\n \* \n \* Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n \* You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n \* The last char sequence may have fewer characters than the given [size].\n \* \n \* @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @sample samples.text.Strings.chunkedTransformToSequence\n \* \n \* @SinceKotlin("1.2")\n public fun <R> CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {\n return windowedSequence(size, size, partialWindows = true, transform = transform)\n }\n\n\n \* Splits the original char sequence into pair of char sequences,\n \* where \*first\* char sequence contains characters for which [predicate] yielded `true`,\n \* while \*second\* char sequence contains characters for which [predicate] yielded `false`.\n \* \n \* @sample samples.text.Strings.partition\n \* \n \* @SinceKotlin("1.2")\n public inline fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {\n val first = StringBuilder()\n val second = StringBuilder()\n for (element in this) {\n if (predicate(element)) {\n first.append(element)\n } else {\n second.append(element)\n }\n }\n return Pair(first, second)\n }\n\n\n \* Splits the original string into pair of strings,\n \* where \*first\* string contains characters for which [predicate] yielded `true`,\n \* while \*second\* string contains characters for which [predicate] yielded `false`.\n \* \n \* @sample samples.text.Strings.partition\n \* \n \* @SinceKotlin("1.2")\n public inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {\n val first = StringBuilder()\n val second = StringBuilder()\n for (element in this) {\n if (predicate(element)) {\n first.append(element)\n }\n else {\n second.append(element)\n }\n }\n return Pair(first.toString(), second.toString())\n }\n\n\n \* Returns a list of snapshots of the window of the given [size]\n \* sliding along this char sequence with the given [step], where each\n \* snapshot is a string.\n \* \n \* Several last strings may have fewer characters than the given [size].\n \* \n \* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @param size the number of elements to take in each window\n \* @param step the number of elements to move the window forward by on an each step, by default 1\n \* @param partialWindows controls whether or not to keep partial windows in the end if any,\n \* by default `false` which means partial windows won't be preserved\n \* \n \* @sample samples.collections.Sequences.Transformations.takeWindows\n \* \n \* @SinceKotlin("1.2")\n public fun CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {\n return windowed(size, step, partialWindows) { it.toString() }\n }\n\n\n \* Returns a list of results of applying the given [transform] function to\n \* an each char sequence representing a view over the window of the given [size]\n \* sliding along this char sequence with the given [step].\n \* \n \* Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n \* You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n \* Several last char sequences may have fewer characters than the given [size].\n \* \n \* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @param size the number of elements to take in each window\n \* @param step the number of

elements to move the window forward by on an each step, by default 1\n \* @param partialWindows controls whether or not to keep partial windows in the end if any,\n \* by default `false` which means partial windows won't be preserved\n \* \n \* @sample samples.collections.Sequences.Transformations.averageWindows\n \*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): List<R> {\n checkWindowSizeStep(size, step)\n val thisSize = this.length\n val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result = ArrayList<R>(resultCapacity)\n var index = 0\n while (index in 0 until thisSize) {\n val end = index + size\n val coercedEnd = if (end < 0 || end > thisSize) { if (partialWindows) thisSize else break } else end\n result.add(transform(subSequence(index, coercedEnd)))\n index += step\n }\n return result\n}\n\n/\*\*\n \* Returns a sequence of snapshots of the window of the given [size]\n \* sliding along this char sequence with the given [step], where each\n \* snapshot is a string.\n \* \n \* Several last strings may have fewer characters than the given [size].\n \* \n \* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @param size the number of elements to take in each window\n \* @param step the number of elements to move the window forward by on an each step, by default 1\n \* @param partialWindows controls whether or not to keep partial windows in the end if any,\n \* by default `false` which means partial windows won't be preserved\n \* \n \* @sample samples.collections.Sequences.Transformations.takeWindows\n \*\n@SinceKotlin("1.2")\npublic fun CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<String> {\n return windowedSequence(size, step, partialWindows) { it.toString() }\n}\n\n/\*\*\n \* Returns a sequence of results of applying the given [transform] function to\n \* an each char sequence representing a view over the window of the given [size]\n \* sliding along this char sequence with the given [step].\n \* \n \* Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n \* You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n \* \n \* Several last char sequences may have fewer characters than the given [size].\n \* \n \* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n \* \n \* @param size the number of elements to take in each window\n \* @param step the number of elements to move the window forward by on an each step, by default 1\n \* @param partialWindows controls whether or not to keep partial windows in the end if any,\n \* by default `false` which means partial windows won't be preserved\n \* \n \* @sample samples.collections.Sequences.Transformations.averageWindows\n \*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {\n checkWindowSizeStep(size, step)\n val windows = (if (partialWindows) indices else 0 until length - size + 1) step step\n return windows.asSequence().map { index ->\n val end = index + size\n val coercedEnd = if (end < 0 || end > length) length else end\n transform(subSequence(index, coercedEnd))\n }\n}\n\n/\*\*\n \* Returns a list of pairs built from the characters of `this` and the [other] char sequences with the same index\n \* The returned list has length of the shortest char sequence.\n \* \n \* @sample samples.text.Strings.zip\n \*\npublic infix fun CharSequence.zip(other: CharSequence): List<Pair<Char, Char>> {\n return zip(other) { c1, c2 -> c1 to c2 }\n}\n\n/\*\*\n \* Returns a list of values built from the characters of `this` and the [other] char sequences with the same index\n \* using the provided [transform] function applied to each pair of characters.\n \* \n \* The returned list has length of the shortest char sequence.\n \* \n \* @sample samples.text.Strings.zipWithTransform\n \*\npublic inline fun <V> CharSequence.zip(other: CharSequence, transform: (a: Char, b: Char) -> V): List<V> {\n val length = minOf(this.length, other.length)\n val list = ArrayList<V>(length)\n for (i in 0 until length) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/\*\*\n \* Returns a list of pairs of each two adjacent characters in this char sequence.\n \* \n \* The returned list is empty if this char sequence contains less than two characters.\n \* \n \* @sample samples.collections.Collections.Transformations.zipWithNext\n \*\n@SinceKotlin("1.2")\npublic fun CharSequence.zipWithNext(): List<Pair<Char, Char>> {\n return zipWithNext { a, b -> a to b }\n}\n\n/\*\*\n

Returns a list containing the results of applying the given [transform] function to an each pair of two adjacent characters in this char sequence.

The returned list is empty if this char sequence contains less than two characters.

```

@sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas
*/n@SinceKotlin("1.2")\npublic inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R):
List<R> {
 val size =
 length - 1
 if (size < 1) return emptyList()
 val result = ArrayList<R>(size)
 for (index in 0 until size) {
 result.add(transform(this[index], this[index + 1]))
 }
 return result
}
*/n
*/n * Creates an [Iterable] instance that wraps the original char sequence returning its characters when being iterated.
*/n\npublic fun CharSequence.asIterable(): Iterable<Char> {
 if (this is String && isEmpty()) return emptyList()
 return Iterable { this.iterator() }
}
*/n
*/n * Creates a [Sequence] instance that wraps the original char sequence returning its characters when being iterated.
*/n\npublic fun CharSequence.asSequence(): Sequence<Char> {
 if (this is String && isEmpty()) return emptySequence()
 return Sequence { this.iterator() }
}
*/n
*/n
*/n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
*/n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/n
*/n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\nimport kotlin.contracts.contract\nimport kotlin.jvm.JvmName\n\n*/n
*/n * Returns a copy of this string converted to upper case using the rules of the default locale.
*/n\n@Deprecated("Use uppercase() instead.", ReplaceWith("uppercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.toUpperCase(): String\n\n*/n
*/n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.
/n\n/n * This function supports one-to-many and many-to-one character mapping,
*/n * thus the length of the returned string can be different from the length of the original string.
/n\n/n *
@sample samples.text.Strings.toUpperCase\n\n
*/n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.toUpperCase(): String\n\n*/n
*/n * Returns a copy of this string converted to lower case using the rules of the default locale.
*/n\n@Deprecated("Use lowercase() instead.", ReplaceWith("lowercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.toLowerCase(): String\n\n*/n
*/n * Returns a copy of this string converted to lower case using Unicode mapping rules of the invariant locale.
/n\n/n * This function supports one-to-many and many-to-one character mapping,
*/n * thus the length of the returned string can be different from the length of the original string.
/n\n/n *
@sample samples.text.Strings.toLowerCase\n\n
*/n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.toLowerCase(): String\n\n*/n
*/n * Returns a copy of this string having its first letter titlecased using the rules of the default locale,
*/n * or the original string if it's empty or already starts with a title case letter.
/n\n/n * The title case of a character is usually the same as its upper case with several exceptions.
*/n * The particular list of characters with the special title case form depends on the underlying platform.
/n\n/n *
@sample samples.text.Strings.capitalize\n\n
*/n\n@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase())
it.titlecase() else it.toString() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.capitalize(): String\n\n*/n
*/n * Returns a copy of this string having its first letter lowercased using the rules of the default locale,
*/n * or the original string if it's empty or already starts with a lower case letter.
/n\n/n *
@sample samples.text.Strings.decapitalize\n\n
*/n\n@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { it.lowercase() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
expect fun String.decapitalize(): String\n\n*/n
*/n * Returns a sub sequence of this char sequence having leading and trailing characters matching the [predicate] removed.
*/n\npublic inline fun CharSequence.trim(predicate: (Char) -> Boolean): CharSequence {
 var startIndex = 0
 var endIndex = length - 1
 var startFound = false
 while (startIndex <= endIndex) {
 val index = if (!startFound) startIndex else endIndex
 val match = predicate(this[index])
 if (!startFound) {
 if (!match) startFound = true
 } else {
 startIndex += 1
 } else {
 if (!match) break
 } else {
 endIndex -= 1
 }
 }
 return subSequence(startIndex, endIndex + 1)
}
*/n
*/n
*/n * Returns a string having leading and

```

trailing characters matching the [predicate] removed.

```

\n *^public inline fun String.trim(predicate: (Char) ->
Boolean): String =\n (this as CharSequence).trim(predicate).toString()\n\n/**\n * Returns a sub sequence of this
char sequence having leading characters matching the [predicate] removed.\n *^public inline fun
CharSequence.trimStart(predicate: (Char)
-> Boolean): CharSequence {\n for (index in this.indices)\n if (!predicate(this[index]))\n return
subSequence(index, length)\n return ""\n}\n\n/**\n * Returns a string having leading characters matching the
[predicate] removed.\n *^public inline fun String.trimStart(predicate: (Char) -> Boolean): String =\n (this as
CharSequence).trimStart(predicate).toString()\n\n/**\n * Returns a sub sequence of this char sequence having
trailing characters matching the [predicate] removed.\n *^public inline fun CharSequence.trimEnd(predicate:
(Char) -> Boolean): CharSequence {\n for (index in this.indices.reversed())\n if (!predicate(this[index]))\n
return subSequence(0, index + 1)\n return ""\n}\n\n/**\n * Returns a string having trailing characters
matching the [predicate] removed.\n *^public inline fun String.trimEnd(predicate: (Char) -> Boolean): String =\n
(this as CharSequence).trimEnd(predicate).toString()\n\n/**\n * Returns
a sub sequence of this char sequence having leading and trailing characters from the [chars] array removed.\n
*^public fun CharSequence.trim(vararg chars: Char): CharSequence = trim { it in chars }\n\n/**\n * Returns a
string having leading and trailing characters from the [chars] array removed.\n *^public fun String.trim(vararg
chars: Char): String = trim { it in chars }\n\n/**\n * Returns a sub sequence of this char sequence having leading
characters from the [chars] array removed.\n *^public fun CharSequence.trimStart(vararg chars: Char):
CharSequence = trimStart { it in chars }\n\n/**\n * Returns a string having leading characters from the [chars]
array removed.\n *^public fun String.trimStart(vararg chars: Char): String = trimStart { it in chars }\n\n/**\n *
Returns a sub sequence of this char sequence having trailing characters from the [chars] array removed.\n
*^public fun
CharSequence.trimEnd(vararg chars: Char): CharSequence = trimEnd { it in chars }\n\n/**\n * Returns
a string having trailing characters from the [chars] array removed.\n *^public fun String.trimEnd(vararg chars:
Char): String = trimEnd { it in chars }\n\n/**\n * Returns a sub sequence of this char sequence having leading
and trailing whitespace removed.\n *^public fun CharSequence.trim(): CharSequence =
trim(Char::isWhitespace)\n\n/**\n * Returns a string having leading and trailing whitespace removed.\n
*^@kotlin.internal.InlineOnly\npublic inline fun String.trim(): String = (this as
CharSequence).trim().toString()\n\n/**\n * Returns a sub sequence of this char sequence having leading whitespace
removed.\n *^public fun CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)\n\n/**\n *
Returns a string having leading whitespace removed.\n *^@kotlin.internal.InlineOnly\npublic inline fun
String.trimStart(): String = (this as CharSequence).trimStart().toString()\n\n/**\n * Returns a sub sequence of this
char sequence having trailing whitespace removed.\n *^public
fun CharSequence.trimEnd(): CharSequence = trimEnd(Char::isWhitespace)\n\n/**\n * Returns a string having
trailing whitespace removed.\n *^@kotlin.internal.InlineOnly\npublic inline fun String.trimEnd(): String = (this
as CharSequence).trimEnd().toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded
at the beginning\n * to the specified [length] with the specified character or space.\n * @param length the desired
string length.\n * @param padChar the character to pad string with, if it has length less than the [length]
specified. Space is used by default.\n * @return Returns a char sequence of length at least [length] consisting
of `this` char sequence prepended with [padChar] as many times\n * as are necessary to reach that length.\n *
@sample samples.text.Strings.padStart\n *^public fun CharSequence.padStart(length: Int, padChar: Char = ' '):
CharSequence {\n if (length < 0)\n throw IllegalArgumentException("\Desired length
$length is less than zero.")\n if (length <= this.length)\n return this.subSequence(0, this.length)\n val sb =
StringBuilder(length)\n for (i in 1..(length - this.length))\n sb.append(padChar)\n sb.append(this)\n return
sb\n}\n\n/**\n * Pads the string to the specified [length] at the beginning with the specified character or space.\n
* @param length the desired string length.\n * @param padChar the character to pad string with, if it has length
less than the [length] specified. Space is used by default.\n * @return Returns a string of length at least
[length] consisting of `this` string prepended with [padChar] as many times\n * as are necessary to reach that
length.\n * @sample samples.text.Strings.padStart\n *^public fun String.padStart(length: Int, padChar: Char = ' '):
String =\n

```



```

(this as CharSequence).padStart(length, padChar).toString()\n\n/**\n * Returns a char sequence with content of this
char sequence padded at the end\n * to the specified
[length] with the specified character or space.\n *\n * @param length the desired string length.\n * @param
padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.\n *
@return Returns a char sequence of length at least [length] consisting of `this` char sequence appended with
[padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padEnd\n
*/\npublic fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {\n if (length < 0)\n throw IllegalArgumentException("Desired length $length is less than zero.")\n if (length <= this.length)\n return this.subSequence(0, this.length)\n val sb = StringBuilder(length)\n sb.append(this)\n for (i in
1..(length - this.length))\n sb.append(padChar)\n return sb\n}\n\n/**\n * Pads the string to the specified
[length] at the end with the specified character or space.\n *\n * @param length
the desired string length.\n * @param padChar the character to pad string with, if it has length less than the [length]
specified. Space is used by default.\n * @return Returns a string of length at least [length] consisting of `this` string
appended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample
samples.text.Strings.padEnd\n */\npublic fun String.padEnd(length: Int, padChar: Char = ' '): String =\n (this as
CharSequence).padEnd(length, padChar).toString()\n\n/**\n * Returns `true` if this nullable char sequence is either
`null` or empty.\n *\n * @sample samples.text.Strings.stringIsNullOrEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence?.isNullOrEmpty(): Boolean {\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n return this == null || this.length == 0\n}\n\n/**\n * Returns `true` if this char sequence is empty (contains no characters).\n *\n * @sample
samples.text.Strings.stringIsEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isEmpty(): Boolean = length == 0\n\n/**\n * Returns `true` if this char sequence is not empty.\n *\n * @sample samples.text.Strings.stringIsNotEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotEmpty(): Boolean = length > 0\n\n//
implemented differently in JVM and JS\n//public fun String.isBlank(): Boolean = length() == 0 || all {
it.isWhitespace() }\n\n/**\n * Returns `true` if this char sequence is not empty and contains some characters
except of whitespace characters.\n *\n * @sample samples.text.Strings.stringIsNotBlank\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotBlank(): Boolean = !isBlank()\n\n/**\n * Returns `true` if this nullable char sequence is either `null` or empty or consists solely of whitespace characters.\n
*\n * @sample samples.text.Strings.stringIsNullOrBlank\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence?.isNullOrBlank(): Boolean
{\n contract {\n returns(false) implies (this@isNullOrBlank != null)\n }\n return this == null ||
this.isBlank()\n}\n\n/**\n * Iterator for characters of the given char sequence.\n */\npublic operator fun
CharSequence.iterator(): CharIterator = object : CharIterator() {\n private var index = 0\n public override fun
nextChar(): Char = get(index++)\n public override fun hasNext(): Boolean = index < length\n}\n\n/** Returns
the string if it is not `null`, or the empty string otherwise. */\n@kotlin.internal.InlineOnly\npublic inline fun
String?.orEmpty(): String = this ?: ""\n\n/**\n * Returns this char sequence if it's not empty\n * or the result of
calling [defaultValue] function if the char sequence is empty.\n *\n * @sample samples.text.Strings.stringIfEmpty\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifEmpty(defaultValue: () ->
R): R where C : CharSequence, C : R =\n if (isEmpty()) defaultValue() else
this\n\n/**\n * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,\n *
or the result of calling [defaultValue] function otherwise.\n *\n * @sample samples.text.Strings.stringIfNotBlank\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifBlank(defaultValue: () ->
R): R where C : CharSequence, C : R =\n if (isBlank()) defaultValue() else this\n\n/**\n * Returns the range of valid
character indices for this char sequence.\n */\npublic val CharSequence.indices: IntRange\n get() = 0..length -
1\n\n/**\n * Returns the index of the last character in the char sequence or -1 if it is empty.\n */\npublic val
CharSequence.lastIndex: Int\n get() = this.length - 1\n\n/**\n * Returns `true` if this CharSequence has Unicode
surrogate pair at the specified [index].\n */\npublic fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {\n

```

```

return index in 0..length - 2\n && this[index].isHighSurrogate()\n
 && this[index + 1].isLowSurrogate()\n}\n\n/**\n * Returns a substring specified by the given [range] of
indices.\n */\npublic fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive +
1)\n\n/**\n * Returns a subsequence of this char sequence specified by the given [range] of indices.\n */\npublic fun
CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive +
1)\n\n/**\n * Returns a subsequence of this char sequence.\n */\n * This extension is chosen only for invocation with
old-named parameters.\n * Replace parameter names with the same as those of [CharSequence.subSequence].\n
*/\n@kotlin.internal.InlineOnly\n@Suppress("\u0027EXTENSION_SHADOWED_BY_MEMBER\u0027") // false
warning\n@Deprecated("\u0027Use parameters named startIndex and endIndex.\u0027", ReplaceWith("\u0027subSequence(startIndex
= start, endIndex = end)\u0027"))\npublic inline fun String.subSequence(start: Int, end: Int): CharSequence =
subSequence(start, end)\n\n/**\n
 * Returns a substring of chars from a range of this char sequence starting at the [startIndex] and ending right before
the [endIndex].\n */\n * @param startIndex the start index (inclusive).\n * @param endIndex the end index
(exclusive). If not specified, the length of the char sequence is used.\n */\n@kotlin.internal.InlineOnly\npublic inline
fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = subSequence(startIndex,
endIndex).toString()\n\n/**\n * Returns a substring of chars at indices from the specified [range] of this char
sequence.\n */\npublic fun CharSequence.substring(range: IntRange): String = subSequence(range.start,
range.endInclusive + 1).toString()\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the
string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n
*/\npublic fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String
{\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(0,
index)\n}\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain
the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun
String.substringBefore(delimiter: String, missingDelimiterValue: String = this): String {\n val index =
indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns
a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns
[missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringAfter(delimiter:
Char, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1)
missingDelimiterValue else substring(index + 1, length)\n}\n\n/**\n * Returns a substring after
the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue]
which defaults to the original string.\n */\npublic fun String.substringAfter(delimiter: String,
missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1)
missingDelimiterValue else substring(index + delimiter.length, length)\n}\n\n/**\n * Returns a substring before the
last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which
defaults to the original string.\n */\npublic fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue:
String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else
substring(0, index)\n}\n\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string does
not contain the delimiter, returns [missingDelimiterValue] which defaults to
the original string.\n */\npublic fun String.substringBeforeLast(delimiter: String, missingDelimiterValue: String =
this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else
substring(0, index)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the string does
not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun
String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this): String {\n val index =
lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(index + 1,
length)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the string does not contain
the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun
String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {\n val index

```

```

= lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(index + delimiter.length,
length)\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the given range\n * is replaced with the [replacement] char sequence.\n * @param startIndex the index of the first character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the string.\n */\npublic fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence):
CharSequence {\n if (endIndex < startIndex)\n throw IndexOutOfBoundsException("\nEnd index ($endIndex)
is less than start index ($startIndex).")\n val sb = StringBuilder()\n sb.appendRange(this, 0, startIndex)\n sb.append(replacement)\n sb.appendRange(this, endIndex, length)\n return sb\n}\n\n/**\n * Replaces the part of
the string at the given range with the [replacement] char sequence.\n * @param startIndex the
index of the first character to be replaced.\n * @param endIndex the index of the first character after the
replacement to keep in the string.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
String.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): String =\n (this as
CharSequence).replaceRange(startIndex, endIndex, replacement).toString()\n\n/**\n * Returns a char sequence with
content of this char sequence where its part at the given [range]\n * is replaced with the [replacement] char
sequence.\n * @param range The end index of the [range] is included in the part to be replaced.\n */\npublic fun
CharSequence.replaceRange(range: IntRange, replacement: CharSequence): CharSequence =\n replaceRange(range.start, range.endInclusive + 1, replacement)\n\n/**\n * Replace the part of string at the given
[range] with the [replacement] string.\n * @param range The end index of the [range] is included in the part to be replaced.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
String.replaceRange(range: IntRange, replacement: CharSequence): String =\n (this as
CharSequence).replaceRange(range, replacement).toString()\n\n/**\n * Returns a char sequence with content of this
char sequence where its part at the given range is removed.\n * @param startIndex the index of the first
character to be removed.\n * @param endIndex the index of the first character after the removed part to keep in the
string.\n * @param [endIndex] is not included in the removed part.\n */\npublic fun
CharSequence.removeRange(startIndex: Int, endIndex: Int): CharSequence {\n if (endIndex < startIndex)\n throw IndexOutOfBoundsException("\nEnd index ($endIndex) is less than start index ($startIndex).")\n if
(endIndex == startIndex)\n return this.subSequence(0, length)\n val sb = StringBuilder(length - (endIndex -
startIndex))\n sb.appendRange(this, 0, startIndex)\n sb.appendRange(this, endIndex, length)\n return
sb\n}\n\n/**\n * Removes the part of a
string at a given range.\n * @param startIndex the index of the first character to be removed.\n * @param endIndex
the index of the first character after the removed part to keep in the string.\n * @param [endIndex] is not included in the
removed part.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(startIndex: Int, endIndex:
Int): String =\n (this as CharSequence).removeRange(startIndex, endIndex).toString()\n\n/**\n * Returns a char
sequence with content of this char sequence where its part at the given [range] is removed.\n * @param range The end index of
the [range] is included in the removed part.\n */\npublic fun CharSequence.removeRange(range: IntRange):
CharSequence =\n removeRange(range.start, range.endInclusive + 1)\n\n/**\n * Removes the part of a string at the
given [range].\n * @param range The end index of the [range] is included in the removed part.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(range: IntRange): String =\n (this as
CharSequence).removeRange(range).toString()\n\n/**\n * If this char sequence starts with the given [prefix], returns a new char sequence\n * with the prefix removed.
Otherwise, returns a new char sequence with the same characters.\n */\npublic fun
CharSequence.removePrefix(prefix: CharSequence): CharSequence {\n if (startsWith(prefix)) {\n return
subSequence(prefix.length, length)\n }\n return subSequence(0, length)\n}\n\n/**\n * If this string starts with the
given [prefix], returns a copy of this string\n * with the prefix removed. Otherwise, returns this string.\n */\npublic
fun String.removePrefix(prefix: CharSequence): String {\n if (startsWith(prefix)) {\n return
substring(prefix.length)\n }\n return this\n}\n\n/**\n * If this char sequence ends with the given [suffix], returns
a new char sequence\n * with the suffix removed. Otherwise, returns a new char sequence with the same
characters.\n */\npublic fun CharSequence.removeSuffix(suffix: CharSequence):

```

```

CharSequence {\n if (endsWith(suffix)) {\n return subSequence(0, length - suffix.length)\n }\n return
subSequence(0, length)\n}\n\n/**\n * If this string ends with the given [suffix], returns a copy of this string\n * with
the suffix removed. Otherwise, returns this string.\n */\npublic fun String.removeSuffix(suffix: CharSequence):
String {\n if (endsWith(suffix)) {\n return substring(0, length - suffix.length)\n }\n return this\n}\n\n/**\n *
When this char sequence starts with the given [prefix] and ends with the given [suffix],\n * returns a new char
sequence having both the given [prefix] and [suffix] removed.\n * Otherwise returns a new char sequence with the
same characters.\n */\npublic fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence):
CharSequence {\n if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n
 return subSequence(prefix.length, length - suffix.length)\n }\n return subSequence(0, length)\n}\n\n/**\n * Removes from a string both the given [prefix] and [suffix] if and
only if\n * it starts with the [prefix] and ends with the [suffix].\n * Otherwise returns this string unchanged.\n */\npublic fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {\n if ((length >=
prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n return substring(prefix.length,
length - suffix.length)\n }\n return this\n}\n\n/**\n * When this char sequence starts with and ends with the
given [delimiter],\n * returns a new char sequence having this [delimiter] removed both from the start and end.\n *
Otherwise returns a new char sequence with the same characters.\n */\npublic fun
CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence = removeSurrounding(delimiter,
delimiter)\n}\n\n/**\n * Removes the given [delimiter] string from both the start and the end of this string\n * if and
only
if it starts with and ends with the [delimiter].\n * Otherwise returns this string unchanged.\n */\npublic fun
String.removeSurrounding(delimiter: CharSequence): String = removeSurrounding(delimiter, delimiter)\n}\n\n/**\n *
Replace part of string before the first occurrence of given delimiter with the [replacement] string.\n * If the string
does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic
fun String.replaceBefore(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n
 val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index,
replacement)\n}\n\n/**\n * Replace part of string before the first occurrence of given delimiter with the
[replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults
to the original string.\n */\npublic fun String.replaceBefore(delimiter: String, replacement:
String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1)
missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n/**\n * Replace part of string after the first
occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns
[missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfter(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {\n val index = indexOf(delimiter)\n return if
(index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n\n/**\n * Replace part
of string after the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain
the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun
String.replaceAfter(delimiter: String, replacement: String, missingDelimiterValue:
String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else
replaceRange(index + delimiter.length, length, replacement)\n}\n\n/**\n * Replace part of string after the last
occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns
[missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfterLast(delimiter:
String, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n
 return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length,
replacement)\n}\n\n/**\n * Replace part of string after the last occurrence of given delimiter with the [replacement]
string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original
string.\n */\npublic fun String.replaceAfterLast(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n
 return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n}\n\n/**\n *

```

Replace part of string before the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceBeforeLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

```

Replace part of string before the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceBeforeLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
 val index = lastIndexOf(delimiter)
 return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

```

public fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific

public fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the given [replacement]. The [replacement] can consist of any combination of literal text and \$-substitutions. To treat the replacement string literally escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)

```

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a replacement for that match.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> CharSequence): String =
 regex.replace(this, transform)

```

Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.

@param replacement A replacement expression that can include substitutions. See [Regex.replaceFirst] for details.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String = regex.replaceFirst(this, replacement)

```

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

@param transform function that takes the first character and returns the result of the transform applied to the character.

```

@sample samples.text.Strings.replaceFirstChar

```

```

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithChar")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> Char): String {
 return if (isEmpty()) transform(this[0]) + substring(1) else this
}

```

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

@param transform function that takes the first character and returns the result of the transform applied to the character.

```

@sample samples.text.Strings.replaceFirstChar

```

```

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithCharSequence")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> CharSequence): String {
 return if (isEmpty()) transform(this[0]).toString() + substring(1) else this
}

```

Returns true if this char sequence matches the given regular expression.

```

@kotlin.internal.InlineOnly
public infix fun CharSequence.matches(regex: Regex): Boolean = regex.matches(this)

```

Implementation of [regionMatches] for CharSequences.

Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.

```

internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {
 if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset > other.length - length))
 return false
 for (index in 0 until length)
 if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase))
 return false
 return true
}

```

```

false\n } \n return true\n}\n\n/**\n * Returns `true` if this char sequence starts with the specified character.\n *\npublic fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n this.length > 0 && this[0].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence ends with the specified character.\n *\npublic fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n this.length > 0 && this[lastIndex].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence starts with the specified prefix.\n *\npublic fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && prefix is String)\n return this.startsWith(prefix)\n else\n return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this char sequence starting at the specified offset [startIndex] starts with the specified prefix.\n *\npublic fun CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && prefix is String)\n return this.startsWith(prefix, startIndex)\n else\n return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this char sequence ends with the specified suffix.\n *\npublic fun CharSequence.endsWith(suffix: CharSequence, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && suffix is String)\n return this.endsWith(suffix)\n else\n return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)\n}\n\n\n//\n common prefix and suffix\n\n/**\n * Returns the longest string `prefix` such that this char sequence and [other] char sequence both start with this prefix,\n * taking care not to split surrogate pairs.\n * If this and [other] have no common prefix, returns the empty string.\n\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonPrefixWith\n *\npublic fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n val shortestLength = minOf(this.length, other.length)\n var i = 0\n while (i < shortestLength && this[i].equals(other[i], ignoreCase = ignoreCase)) {\n i++\n }\n if (this.hasSurrogatePairAt(i - 1) || other.hasSurrogatePairAt(i - 1)) {\n i--\n }\n return subSequence(0, i).toString()\n}\n\n/**\n * Returns the longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,\n * taking care not to split surrogate pairs.\n * If this and [other] have no common suffix, returns the empty string.\n\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonSuffixWith\n *\npublic fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n val thisLength = this.length\n val otherLength = other.length\n val shortestLength = minOf(thisLength, otherLength)\n var i = 0\n while (i < shortestLength && this[thisLength - i - 1].equals(other[otherLength - i - 1], ignoreCase = ignoreCase)) {\n i++\n }\n if (this.hasSurrogatePairAt(thisLength - i - 1) || other.hasSurrogatePairAt(otherLength - i - 1)) {\n i--\n }\n return subSequence(thisLength - i, thisLength).toString()\n}\n\n\n// indexOfAny()\n\n/**\n * Finds the index of the first occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the first occurrence of matched character from [chars] or -1 if none of [chars] are found.\n *\npublic fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n if (!ignoreCase && chars.size == 1 && this is String) {\n val char = chars.single()\n return nativeIndexOf(char, startIndex)\n }\n\n for (index in startIndex.coerceAtLeast(0)..lastIndex) {\n val charAtIndex = get(index)\n if (chars.any { it.equals(charAtIndex, ignoreCase) })\n return index\n }\n return -1\n}\n\n/**\n * Finds the index of the last occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.\n *\npublic fun CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n if (!ignoreCase && chars.size == 1 && this is String) {\n val char = chars.single()\n return

```

```

nativeLastIndexOf(char, startIndex)\n }\n\n for (index in startIndex.coerceAtMost(lastIndex) downTo 0) {\n val charAtIndex = get(index)\n if (chars.any { it.equals(charAtIndex, ignoreCase) })\n return index\n }\n return -1\n}\n\nprivate fun CharSequence.indexOf(other: CharSequence, startIndex: Int, endIndex: Int, ignoreCase: Boolean, last: Boolean = false): Int {\n val indices = if (!last)\n startIndex.coerceAtLeast(0)..endIndex.coerceAtMost(length)\n else\n startIndex.coerceAtMost(lastIndex) downTo endIndex.coerceAtLeast(0)\n\n if (this is String && other is String) { // smart cast\n for (index in indices) {\n if (other.regionMatches(0, this, index, other.length, ignoreCase))\n return index\n }\n } else {\n for (index in indices) {\n if (other.regionMatchesImpl(0, this, index, other.length, ignoreCase))\n return index\n }\n }\n return -1\n}\n\nprivate fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int, ignoreCase: Boolean, last: Boolean): Pair<Int, String>? {\n if (!ignoreCase && strings.size == 1) {\n val string = strings.single()\n val index = if (!last) indexOf(string, startIndex) else lastIndexOf(string, startIndex)\n return if (index < 0) null else index to string\n }\n\n val indices = if (!last) startIndex.coerceAtLeast(0)..length else startIndex.coerceAtMost(lastIndex) downTo 0\n\n if (this is String) {\n for (index in indices) {\n val matchingString = strings.firstOrNull { it.regionMatches(0, this, index, it.length, ignoreCase) }\n if (matchingString != null)\n return index to matchingString\n }\n } else {\n for (index in indices) {\n val matchingString = strings.firstOrNull { it.regionMatchesImpl(0, this, index, it.length, ignoreCase) }\n if (matchingString != null)\n return index to matchingString\n }\n }\n\n return null\n}\n\n/**\n * Finds the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the first occurrence of matched string from [strings] and the string matched\n * or `null` if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Pair<Int, String>? =\n findAnyOf(strings, startIndex, ignoreCase, last = false)\n\n/**\n * Finds the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the last occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =\n findAnyOf(strings, startIndex, ignoreCase, last = true)\n\n/**\n * Finds the index of the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase, last = false)?.first ?: -1\n\n/**\n * Finds the index of the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of matched string from [strings] or

```

```

-1 if none of [strings] are found.\n *\n * To avoid ambiguous
results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the
beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that
position.\n *\npublic fun CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex,
ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1\n\n//
indexOf\n\n/**\n * Returns the index within this string of the first occurrence of the specified character, starting
from the specified [startIndex].\n *\n * @param ignoreCase `true` to ignore character case when matching a
character. By default `false`.\n * @return An index of the first occurrence of [char] or -1 if none is found.\n
*\npublic fun CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return
if (ignoreCase || this !is String)\n indexOfAny(charArrayOf(char), startIndex,
ignoreCase)\n else\n nativeIndexOf(char, startIndex)\n}\n\n/**\n * Returns the index within this char
sequence of the first occurrence of the specified [string],\n * starting from the specified [startIndex].\n *\n * @param
ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the
first occurrence of [string] or -1 if none is found.\n * @sample samples.text.Strings.indexOf\n *\npublic fun
CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return if
(ignoreCase || this !is String)\n indexOf(string, startIndex, length, ignoreCase)\n else\n nativeIndexOf(string, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of
the specified character,\n * starting from the specified [startIndex].\n *\n * @param startIndex The index of
character to start searching at. The search proceeds backward toward the beginning of the string.\n
* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An
index of the last occurrence of [char] or -1 if none is found.\n *\npublic fun CharSequence.lastIndexOf(char: Char,
startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n
 lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)\n else\n nativeLastIndexOf(char,
startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of the specified
[string],\n * starting from the specified [startIndex].\n *\n * @param startIndex The index of character to start
searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to
ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of
[string] or -1 if none is found.\n *\npublic fun CharSequence.lastIndexOf(string: String,
startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n
 indexOf(string, startIndex, 0, ignoreCase, last = true)\n else\n nativeLastIndexOf(string,
startIndex)\n}\n\n/**\n * Returns `true` if this char sequence contains the specified [other] sequence of characters as
a substring.\n *\n * @param ignoreCase `true` to ignore character case when comparing strings. By default `false`.\n
*\n * @Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic operator fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =\n if (other is String)\n
 indexOf(other, ignoreCase = ignoreCase) >= 0\n else\n indexOf(other, 0, length, ignoreCase) >=
0\n\n\n/**\n * Returns `true` if this char sequence\n contains the specified character [char].\n *\n * @param
ignoreCase `true` to ignore character case when comparing characters. By default `false`.\n
*\n * @Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic
operator fun CharSequence.contains(char: Char, ignoreCase: Boolean = false): Boolean =\n indexOf(char,
ignoreCase = ignoreCase) >= 0\n\n/**\n * Returns `true` if this char sequence contains at least one match of the
specified regular expression [regex].\n *\n * @kotlin.internal.InlineOnly\npublic inline operator fun
CharSequence.contains(regex: Regex): Boolean = regex.containsMatchIn(this)\n\n\n//
rangesDelimitedBy\n\n\nprivate class DelimitedRangesSequence(\n private val input: CharSequence,\n private
val startIndex: Int,\n private val limit: Int,\n private val getNextMatch: CharSequence.(currentIndex: Int) ->
Pair<Int, Int>?) : Sequence<IntRange> {\n override fun iterator(): Iterator<IntRange> = object :
Iterator<IntRange> {\n var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n var
currentStartIndex: Int = startIndex.coerceIn(0, input.length)\n var nextSearchIndex: Int = currentStartIndex\n
var nextItem:

```



```

IntRange? = null\n var counter: Int = 0\n private fun calcNext() {\n if (nextSearchIndex < 0) {\n nextState = 0\n nextItem = null\n } else {\n if (limit > 0 && ++counter >= limit ||\n nextSearchIndex > input.length) {\n nextItem = currentStartIndex..input.lastIndex\n nextSearchIndex = -1\n } else {\n val match = input.getNextMatch(nextSearchIndex)\n if (match == null) {\n nextItem = currentStartIndex..input.lastIndex\n nextSearchIndex = -1\n } else {\n val (index, length) = match\n nextItem =\n currentStartIndex until index\n currentStartIndex = index + length\n nextSearchIndex =\n currentStartIndex + if (length == 0) 1 else 0\n }\n }\n nextState = 1\n }\n }\n override fun next(): IntRange {\n if (nextState == -\n 1)\n calcNext()\n if (nextState == 0)\n throw NoSuchElementException()\n val\n result = nextItem as IntRange\n // Clean next to avoid keeping reference on yielded instance\n nextItem = null\n nextState = -1\n return result\n }\n override fun hasNext(): Boolean {\n if (nextState == -1)\n calcNext()\n return nextState == 1\n }\n}\n\n/**\n * Returns a\n * sequence of index ranges of substrings in this char sequence around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more characters to be used as delimiters.\n * @param startIndex The index to start\n * searching delimiters from.\n * No range having its start value less than [startIndex] is returned.\n * [startIndex] is\n * coerced\n * to be non-negative and not greater than length of this string.\n * @param ignoreCase `true` to ignore character case\n * when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return. Zero\n * by default means no limit is set.\n */\nprivate fun CharSequence.rangesDelimitedBy(delimiters: CharArray,\n startIndex: Int = 0, ignoreCase: Boolean = false, limit: Int = 0): Sequence<IntRange> {\n requireNonNegativeLimit(limit)\n return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->\n indexOfAny(delimiters, currentIndex, ignoreCase = ignoreCase).let { if (it < 0) null else it to 1 }\n })\n}\n\n/**\n * Returns a sequence of index ranges of substrings in this char sequence around occurrences of the\n * specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param\n * startIndex The index to start searching delimiters from.\n * No range having its start value less than [startIndex] is\n * returned.\n * [startIndex] is coerced to be non-negative and not greater than length of this string.\n * @param\n * ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The\n * maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results\n * when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of\n * this string, and finds at each position the first element in [delimiters]\n * that matches this string at that position.\n */\nprivate fun CharSequence.rangesDelimitedBy(delimiters: Array<out String>, startIndex: Int = 0, ignoreCase:\n Boolean = false, limit: Int = 0): Sequence<IntRange> {\n requireNonNegativeLimit(limit)\n val delimitersList =\n delimiters.asList()\n return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->\n findAnyOf(delimitersList, currentIndex, ignoreCase = ignoreCase, last = false)?.let { it.first\n to it.second.length }\n })\n}\n\ninternal fun requireNonNegativeLimit(limit: Int) =\n require(limit >= 0) { \"Limit\n * must be non-negative, but was $limit\" }\n\n// split\n\n/**\n * Splits this char sequence to a sequence of strings\n * around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as\n * delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *\n * @param limit The maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid\n * ambiguous results when strings in [delimiters] have characters in common, this method proceeds from\n * the\n * beginning to the end of this string, and finds at each position the first element in [delimiters]\n * that matches this\n * string at that position.\n */\npublic fun CharSequence.splitToSequence(vararg delimiters: String, ignoreCase:\n Boolean = false, limit: Int = 0): Sequence<String> =\n rangesDelimitedBy(delimiters,\n ignoreCase = ignoreCase, limit = limit).map { substring(it) }\n\n/**\n * Splits this char sequence to a list of strings\n * around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as\n * delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *\n * @param limit The maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid

```

ambiguous results when strings in [delimiters] have characters in common, this method proceeds from the beginning to the end of this string, and matches at each position the first element in [delimiters] that is equal to a delimiter in this instance at that position.

```

public fun CharSequence.split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
 if (delimiters.size == 1) {
 val delimiter = delimiters[0]
 if (!delimiter.isEmpty()) {
 return split(delimiter, ignoreCase, limit)
 }
 }
 return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}

```

**Splits this char sequence to a sequence of strings around occurrences of the specified [delimiters].** One or more characters to be used as delimiters. `ignoreCase`true`` to ignore character case when matching a delimiter. By default `false``. `limit`` The maximum number of substrings to return.

```

public fun CharSequence.splitToSequence(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =
 rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }

```

**Splits this char sequence to a list of strings around occurrences of the specified [delimiters].** One or more characters to be used as delimiters. `ignoreCase`true`` to ignore character case when matching a delimiter. By default `false``. `limit`` The maximum number of substrings to return.

```

public fun CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
 if (delimiters.size == 1) {
 return split(delimiters[0].toString(), ignoreCase, limit)
 }
 return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}

```

**Splits this char sequence to a list of strings around occurrences of the specified [delimiter].** This is specialized version of split which receives single non-empty delimiter and offers better performance. `delimiter`` String used as delimiter. `ignoreCase`true`` to ignore character case when matching a delimiter. By default `false``. `limit`` The maximum number of substrings to return.

```

private fun CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {
 requireNonNegativeLimit(limit)
 var currentOffset = 0
 var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
 if (nextIndex == -1 || limit == 1) {
 return listOf(this.toString())
 }
 val isLimited = limit > 0
 val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)
 do {
 result.add(substring(currentOffset, nextIndex))
 currentOffset = nextIndex + delimiter.length
 // Do not search for next occurrence if we're reaching limit
 if (isLimited && result.size == limit - 1) break
 nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
 } while (nextIndex != -1)
 result.add(substring(currentOffset, length))
 return result
}

```

**Splits this char sequence to a list of strings around matches of the given regular expression.** `limit`` Non-negative value specifying the maximum number of substrings to return. Zero by default means no limit is set.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.split(regex: Regex, limit: Int = 0): List<String> = regex.split(this, limit)

```

**Splits this char sequence to a sequence of strings around matches of the given regular expression.** `limit`` Non-negative value specifying the maximum number of substrings to return. Zero by default means no limit is set.

```

@sample samples.text.Strings.splitToSequence

```

```

@SinceKotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun CharSequence.splitToSequence(regex: Regex, limit: Int = 0): Sequence<String> =
 regex.splitToSequence(this, limit)

```

**Splits this char sequence to a sequence of lines delimited by any of the following character sequences: CRLF, LF or CR.** The lines returned do not include terminating line separators.

```

public fun CharSequence.lineSequence(): Sequence<String> = splitToSequence("\\r\\n", "\\n", "\\r")

```

**Splits this char sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR.** The lines returned do not include terminating line separators.

```

public fun CharSequence.lines(): List<String> = lineSequence().toList()

```

**Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],** i.e. both char sequences contain the same number of the same characters in the same order.

```

@sample samples.text.Strings.contentEquals

```

```

@SinceKotlin("1.5")
public expect infix fun

```

```

CharSequence?.contentEquals(other: CharSequence?): Boolean\n\n/**\n * Returns `true` if the contents of this char
sequence are equal to the contents of the specified [other], optionally ignoring case difference.\n *\n * @param
ignoreCase `true` to ignore character case when comparing contents.\n *\n * @sample
samples.text.Strings.contentEquals\n */\n\n@SinceKotlin("1.5")\n\npublic expect fun
CharSequence?.contentEquals(other: CharSequence?,
ignoreCase: Boolean): Boolean\n\ninternal fun CharSequence?.contentEqualsIgnoreCaseImpl(other:
CharSequence?): Boolean {\n if (this is String && other is String) {\n return this.equals(other, ignoreCase =
true)\n }\n if (this === other) return true\n if (this == null || other == null || this.length != other.length) return
false\n for (i in 0 until length) {\n if (!this[i].equals(other[i], ignoreCase = true)) {\n return false\n
 }\n }\n return true\n}\n\ninternal fun CharSequence?.contentEqualsImpl(other: CharSequence?): Boolean {\n
if (this is String && other is String) {\n return this == other\n }\n if (this === other) return true\n if (this
== null || other == null || this.length != other.length) return false\n for (i in 0 until length) {\n if (this[i] !=
other[i]) {\n return false\n }\n }\n return true\n}\n\n/**\n * Returns `true` if the content of this
string
is equal to the word `true`, `false` if it is equal to `false`,\n * and throws an exception otherwise.\n *\n * There is
also a lenient version of the function available on nullable String, [String?.toBoolean].\n * Note that this function is
case-sensitive.\n *\n * @sample samples.text.Strings.toBooleanStrict\n */\n\n@SinceKotlin("1.5")\n\npublic fun
String.toBooleanStrict(): Boolean = when (this) {\n `true` -> true\n `false` -> false\n else -> throw
IllegalArgumentException("The string doesn't represent a boolean value: $this")\n}\n\n/**\n * Returns `true` if the
content of this string is equal to the word `true`, `false` if it is equal to `false`,\n * and `null` otherwise.\n *\n *
There is also a lenient version of the function available on nullable String, [String?.toBoolean].\n * Note that this
function is case-sensitive.\n *\n * @sample samples.text.Strings.toBooleanStrictOrNull\n */\n\n@SinceKotlin("1.5")\n\npublic fun String.toBooleanStrictOrNull(): Boolean?
= when (this) {\n `true` -> true\n `false` -> false\n else -> null\n}, "/*\n * Copyright 2010-2023 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.3")\n\n@ExperimentalUnsignedTypes\n\n@JvmInline\n\npublic value class
UByteArray\n\n@PublishedApi\n\ninternal constructor(@PublishedApi internal val storage: ByteArray) :
Collection<UByte> {\n /** Creates a new array of the specified [size], with all elements initialized to zero. *\n
public constructor(size: Int) : this(ByteArray(size))\n /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n *\n * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n *\n * public operator fun get(index: Int): UByte = storage[index].toUByte()\n *\n *
Sets the element at the given [index] to the given [value]. This method can be called using the index operator.\n *\n
 * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n *
where the behavior is unspecified.\n *\n * public operator fun set(index: Int, value: UByte) {\n
storage[index] = value.toByte()\n }\n /** Returns the number of elements in the array. *\n public override
val size: Int get() = storage.size\n /** Creates an iterator over the elements of the array. *\n public override
operator fun iterator(): kotlin.collections.Iterator<UByte> = Iterator(storage)\n private class Iterator(private val
array: ByteArray) : kotlin.collections.Iterator<UByte> {\n private var index = 0\n override fun hasNext() =
index < array.size\n override fun
next() = if (index < array.size) array[index++].toUByte() else throw NoSuchElementException(index.toString())\n
 }\n override fun contains(element: UByte): Boolean {\n // TODO: Eliminate this check after KT-30016 gets
fixed.\n // Currently JS BE does not generate special bridge method for this method.\n
 @Suppress("USELESS_CAST")\n if ((element as Any?) !is UByte) return false\n return
storage.contains(element.toByte())\n }\n override fun containsAll(elements: Collection<UByte>): Boolean {\n
return (elements as Collection<*>).all { it is UByte && storage.contains(it.toByte()) }\n }\n override fun

```

isEmpty(): Boolean = this.storage.size == 0\n\n\*\*\n \* Creates a new array of the specified [size], where each element is calculated by calling the specified\n \* [init] function.\n \* The function [init] is called for each array element sequentially starting from the first one.\n \* It should return the value for an array element given its index.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray(size: Int, init: (Int) -> UByte): UByteArray {\n return UByteArray(ByteArray(size) { index ->
init(index).toByte()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ubyteArrayOf(vararg elements: UByte): UByteArray = elements\n", /*\n * Copyright 2010-2023 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class UIntArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: IntArray) :
Collection<UInt> {\n\n /** Creates a new array of the specified [size], with all elements initialized
to zero. *\n public constructor(size: Int) : this(IntArray(size))\n\n /**\n * Returns the array element at the
given [index]. This method can be called using the index operator.\n *\n * If the [index] is out of bounds of this
array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n\n *\n public operator fun get(index: Int): UInt = storage[index].toInt()\n\n /**\n * Sets the element at the
given [index] to the given [value]. This method can be called using the index operator.\n *\n * If the [index] is
out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n\n *\n public operator fun set(index: Int, value: UInt) {\n storage[index] = value.toInt()\n }\n\n /** Returns the number of elements in the array. *\n public override val size: Int get() = storage.size\n\n /** Creates an iterator over
the elements of the array. *\n public override operator fun iterator(): kotlin.collections.Iterator<UInt> =
Iterator(storage)\n\n private class Iterator(private val array: IntArray) : kotlin.collections.Iterator<UInt> {\n private var index = 0\n override fun hasNext() = index < array.size\n override fun next() = if (index <
array.size) array[index++].toInt() else throw NoSuchElementException(index.toString())\n }\n\n override fun
contains(element: UInt): Boolean {\n // TODO: Eliminate this check after KT-30016 gets fixed.\n //
Currently JS BE does not generate special bridge method for this method.\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UInt) return false\n return
storage.contains(element.toInt())\n }\n\n override fun containsAll(elements: Collection<UInt>): Boolean {\n return (elements as Collection<*>).all { it is UInt && storage.contains(it.toInt()) }\n }\n\n override fun
isEmpty():
```

Boolean = this.storage.size == 0\n\n\*\*\n \* Creates a new array of the specified [size], where each element is calculated by calling the specified\n \* [init] function.\n \* The function [init] is called for each array element sequentially starting from the first one.\n \* It should return the value for an array element given its index.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray(size: Int, init: (Int) -> UInt): UIntArray {\n return UIntArray(IntArray(size) { index ->
init(index).toInt()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
uintArrayOf(vararg elements: UInt): UIntArray = elements\n", /*\n * Copyright 2010-2023 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class ULongArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: LongArray) :
Collection<ULong> {\n\n /** Creates a new array of the specified [size], with all elements initialized to zero. *\n public constructor(size: Int) : this(LongArray(size))\n\n /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n *\n * If the [index] is out of bounds of this array, throws
```

```

an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n *\n public
operator fun get(index: Int): ULong = storage[index].toULong()\n\n /**\n * Sets the element at the given
[index] to the given [value]. This method can be called using the index operator.\n *\n * If the [index] is out of
bounds of this array, throws an [IndexOutOfBoundsException] except
in Kotlin/JS\n * where the behavior is unspecified.\n *\n public operator fun set(index: Int, value: ULong)
{\n storage[index] = value.toLong()\n }\n\n /** Returns the number of elements in the array. *\n public
override val size: Int get() = storage.size\n\n /** Creates an iterator over the elements of the array. *\n public
override operator fun iterator(): kotlin.collections.Iterator<ULong> = Iterator(storage)\n\n private class
Iterator(private val array: LongArray) : kotlin.collections.Iterator<ULong> {\n private var index = 0\n
override fun hasNext() = index < array.size\n override fun next() = if (index < array.size)
array[index++].toULong() else throw NoSuchElementException(index.toString())\n }\n\n override fun
contains(element: ULong): Boolean {\n // TODO: Eliminate this check after KT-30016 gets fixed.\n //
Currently JS BE does not generate special bridge method for this method.\n
@Suppress(\"USELESS_CAST\")\n
if ((element as Any?) !is ULong) return false\n\n return storage.contains(element.toLong())\n }\n\n
override fun containsAll(elements: Collection<ULong>): Boolean {\n return (elements as Collection<*>).all { it
is ULong && storage.contains(it.toLong()) }\n }\n\n override fun isEmpty(): Boolean = this.storage.size ==
0\n}\n\n/**\n * Creates a new array of the specified [size], where each element is calculated by calling the
specified\n * [init] function.\n *\n * The function [init] is called for each array element sequentially starting from the
first one.\n * It should return the value for an array element given its index.\n
*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray(size: Int, init: (Int) -> ULong): ULongArray {\n return ULongArray(LongArray(size) { index ->
init(index).toLong()
})\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ulongArrayOf(vararg elements: ULong): ULongArray = elements\n\", \"/*\n * Copyright 2010-2023
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class
UShortArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ShortArray) :
Collection<UShort> {\n\n /** Creates a new array of the specified [size], with all elements initialized to zero. *\n
public constructor(size: Int) : this(ShortArray(size))\n\n /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n *\n * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where
the behavior is unspecified.\n *\n public operator fun get(index: Int): UShort = storage[index].toUShort()\n\n
/**\n * Sets the element at the given [index] to the given [value]. This method can be called using the index
operator.\n *\n * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except
in Kotlin/JS\n * where the behavior is unspecified.\n *\n public operator fun set(index: Int, value: UShort)
{\n storage[index] = value.toShort()\n }\n\n /** Returns the number of elements in the array. *\n public
override val size: Int get() = storage.size\n\n /** Creates an iterator over the elements of the array. *\n public
override operator fun iterator(): kotlin.collections.Iterator<UShort> = Iterator(storage)\n\n private class
Iterator(private val array: ShortArray) : kotlin.collections.Iterator<UShort> {\n private var index = 0\n
override fun hasNext() = index < array.size\n
override fun next() = if (index < array.size) array[index++].toUShort() else throw
NoSuchElementException(index.toString())\n }\n\n override fun contains(element: UShort): Boolean {\n //
TODO: Eliminate this check after KT-30016 gets fixed.\n // Currently JS BE does not generate special bridge
method for this method.\n @Suppress(\"USELESS_CAST\")\n if ((element as Any?) !is UShort) return
false\n\n return storage.contains(element.toShort())\n }\n\n override fun containsAll(elements:

```

```

Collection<UShort>: Boolean {
 return (elements as Collection<*>).all { it is UShort &&
storage.contains(it.toShort()) }
}
override fun isEmpty(): Boolean = this.storage.size == 0
}

Creates a new array of the specified [size], where each element is calculated by calling the specified
function. The function [init] is called for each array element sequentially starting from the first one.
It should return
the value for an array element given its index.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UShortArray(size: Int, init: (Int) -> UShort): UShortArray {
 return UShortArray(ShortArray(size) { index ->
init(index).toShort()
})
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ushortArrayOf(vararg elements: UShort): UShortArray = elements

"/

* Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass@file:kotlin.jvm.JvmName("UArraysKt")
@file:kotlin.jvm.JvmPackage("kotlin.collections.unsigned")
package kotlin.collections

NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

import
kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

Returns 1st element*
from the array.
* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
UIntArray.component1(): UInt {
 return get(0)
}

Returns 1st element* from the array.
* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
ULongArray.component1(): ULong {
 return get(0)
}

Returns 1st element* from
the array.
* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the behavior
is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
UByteArray.component1(): UByte {
 return get(0)
}

Returns 1st element* from
the array.
* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
UShortArray.component1(): UShort {
 return get(0)
}

Returns 2nd element* from the array.
* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the
behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
UIntArray.component2(): UInt {
 return get(1)
}

Returns 2nd element* from the array.
* If
the size of this array is
less than 2, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
ULongArray.component2(): ULong {
 return get(1)
}

Returns 2nd element* from the array.
* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the
behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun
UByteArray.component2(): UByte {
 return get(1)
}

Returns 2nd element* from the array.
* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in
Kotlin/JS* where the
behavior is unspecified.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun

```

```

UShortArray.component2(): UShort {
 return
 get(1)}
}
/**
 * Returns 3rd *element* from the array.
 *
 * If the size of this array is less than 3, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UIntArray.component3(): UInt {
 return get(2)}
}
/**
 * Returns 3rd *element* from the array.
 *
 * If
 the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
ULongArray.component3(): ULong {
 return get(2)}
}
/**
 * Returns 3rd *element* from the array.
 *
 *
 If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline operator fun
UByteArray.component3(): UByte {
 return get(2)}
}
/**
 * Returns 3rd *element* from
 the array.
 *
 *
 If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in
 Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UShortArray.component3(): UShort {
 return get(2)}
}
/**
 * Returns 4th *element* from the array.
 *
 *
 If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UIntArray.component4(): UInt {
 return get(3)}
}
/**
 * Returns 4th *element* from the array.
 *
 *
 If the
 size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where
 the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
ULongArray.component4(): ULong {
 return get(3)}
}
/**
 * Returns 4th *element* from the array.
 *
 *
 If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UByteArray.component4(): UByte {
 return get(3)}
}
/**
 * Returns 4th *element* from the array.
 *
 *
 If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UShortArray.component4(): UShort {
 return get(3)}
}
/**
 * Returns 5th *element* from the array.
 *
 *
 If the size
 of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is
 unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline operator fun
UIntArray.component5(): UInt {
 return get(4)}
}
/**
 * Returns 5th *element* from the
 array.
 *
 *
 If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 *
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
ULongArray.component5(): ULong {
 return get(4)}
}
/**
 * Returns 5th *element* from the array.
 *
 *
 If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the
 behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun
UByteArray.component5(): UByte
{
 return get(4)}
}
/**
 * Returns 5th *element* from the array.
 *
 *
 If the size of this array is less than
 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun

```

```

UShortArray.component5(): UShort { \n return get(4)\n}\n\n/**\n * Returns an element at the given [index] or
throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.elementAt(index: Int):
UInt\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.elementAt(index:
Int): ULong\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.elementAt(index: Int):
UByte\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int):
UShort\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
[index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt { \n return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong { \n return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrElse(index:
Int, defaultValue: (Int) -> UByte): UByte { \n return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort { \n return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or
`null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.elementAtOrNull(index: Int): UInt? { \n
 return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrNull(index: Int): ULong? { \n return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrNull(index: Int): UByte? { \n return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample

```



```

samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.elementAtOrNull(index: Int): UShort? {\n return this.getOrNull(index)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.find(predicate: (UInt) -> Boolean): UInt? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.find(predicate: (ULong) -> Boolean): ULong? {\n return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n
* \n * @sample samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.find(predicate: (UByte) -> Boolean): UByte? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns
the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.find(predicate: (UShort) -> Boolean): UShort? {\n return firstOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {\n return lastOrNull(predicate)\n}\n\n/**\n
* Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {\n return lastOrNull(predicate)\n}\n\n/**\n
* Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {\n return lastOrNull(predicate)\n}\n\n/**\n
* Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {\n return
lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.first(): UInt {\n return storage.first().toUInt()\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(): ULong {\n return storage.first().toULong()\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(): UByte {\n return storage.first().toUByte()\n}\n\n/**\n * Returns the first element.\n * \n *
@throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(): UShort {\n return storage.first().toUShort()\n}\n\n/**\n * Returns the first element matching

```

the given [predicate].\n \* @throws [NoSuchElementException] if no such element is found.\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(predicate: (UInt) -> Boolean): UInt {\n for (element in this) if (predicate(element)) return
element\n throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(predicate: (ULong) -> Boolean): ULong {\n for (element in this) if
(predicate(element)) return element\n throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(predicate: (UByte) -> Boolean): UByte {\n for (element in this) if (predicate(element)) return
element\n throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(predicate: (UShort) -> Boolean): UShort {\n for (element in this) if (predicate(element)) return
element\n throw NoSuchElementException("Array contains no
element matching the predicate.")\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.firstOrNull(): UInt? {\n return
if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.firstOrNull(): ULong? {\n
return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.firstOrNull(): UByte? {\n
return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.firstOrNull(): UShort? {\n
return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or
`null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {\n for (element in this) if (predicate(element)) return
element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element
was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {\n for (element in this) if
(predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {\n for (element in this) if (predicate(element))
return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {\n for (element in this) if (predicate(element))
return element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.getOrNull(index: Int, defaultValue: (Int) -> ULong): ULong {\n return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.getOrNull(index: Int, defaultValue: (Int) -> UByte): UByte {\n return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.getOrNull(index: Int, defaultValue: (Int) -> UShort): UShort {\n return if (index >= 0 && index <=
lastIndex) get(index)
else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of
bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.getOrNull(index: Int): UInt? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.getOrNull(index: Int):
ULong? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.getOrNull(index:
Int): UByte? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.getOrNull(index: Int):
UShort? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOf(element: UInt): Int {\n return storage.indexOf(element.toInt())\n}\n\n/**\n * Returns first
index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOf(element: ULong): Int {\n return
storage.indexOf(element.toLong())\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain
element.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UByteArray.indexOf(element: UByte): Int {\n return storage.indexOf(element.toByte())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOf(element: UShort): Int {\n return storage.indexOf(element.toShort())\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {\n return storage.indexOfFirst { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {\n return storage.indexOfFirst {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {\n return storage.indexOfFirst {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {\n return storage.indexOfFirst
{ predicate(it.toUShort()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if
the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {\n return storage.indexOfLast { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {\n return storage.indexOfLast {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfLast(predicate:
(UByte) -> Boolean): Int {\n return storage.indexOfLast { predicate(it.toUByte()) }\n}\n\n/**\n * Returns index
of the last element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {\n return storage.indexOfLast {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(): UInt {\n return storage.last().toUInt()\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.last(): ULong {\n return storage.last().toULong()\n}\n\n/**\n * Returns the last element.\n
* \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(): UByte {\n return storage.last().toUByte()\n}\n\n/**\n * Returns the last element.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(): UShort {\n return storage.last().toUShort()\n}\n\n/**\n * Returns the last element matching
the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(predicate: (UInt) -> Boolean): UInt {\n for (index in this.indices.reversed()) {\n val element =
this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException("Array contains
no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n *
@throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.last(predicate: (ULong) -> Boolean): ULong {\n for (index in this.indices.reversed()) {\n val
element = this[index]\n if (predicate(element)) return element\n }\n throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last

```

```

element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
* \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(predicate: (UByte) -> Boolean): UByte {\n for (index in this.indices.reversed()) {\n val
element = this[index]\n if (predicate(element)) return element\n }\n throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
* \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(predicate: (UShort) -> Boolean): UShort {\n for (index in this.indices.reversed()) {\n val
element = this[index]\n
if (predicate(element)) return element\n }\n throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain
element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.lastIndexOf(element: UInt): Int {\n return storage.lastIndexOf(element.toInt())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.lastIndexOf(element: ULong): Int {\n return storage.lastIndexOf(element.toLong())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.lastIndexOf(element: UByte): Int {\n return storage.lastIndexOf(element.toByte())\n}\n\n/**\n
* Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.lastIndexOf(element: UShort): Int {\n return storage.lastIndexOf(element.toShort())\n}\n\n/**\n
* Returns the last element, or `null` if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.lastOrNull(): UInt? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last
element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.lastOrNull(): ULong? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.lastOrNull(): UByte? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.lastOrNull(): UShort? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given [predicate], or
`null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {\n for (index in this.indices.reversed()) {\n val
element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last
element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {\n for (index in this.indices.reversed()) {\n
val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n

```

```

*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {^n for (index in this.indices.reversed()) {^n
val element = this[index]^n if (predicate(element)) return element^n }^n return null^n}^n/n/**^n * Returns the
last element matching
the given [predicate], or `null` if no such element was found.^n * ^n * @sample
samples.collections.Collections.Elements.last^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {^n for (index in this.indices.reversed()) {^n
val element = this[index]^n if (predicate(element)) return element^n }^n return null^n}^n/n/**^n * Returns a
random element from this array.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UIntArray.random(): UInt {^n return random(Random)^n}^n/n/**^n * Returns a random element from this array.^n
* ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
ULongArray.random(): ULong {^n return random(Random)^n}^n/n/**^n
* Returns a random element from this array.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UByteArray.random(): UByte {^n return random(Random)^n}^n/n/**^n * Returns a random element from this
array.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
UShortArray.random(): UShort {^n return random(Random)^n}^n/n/**^n * Returns a random element from this
array using the specified source of randomness.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UIntArray.random(random: Random): UInt
{^n if (isEmpty())^n throw NoSuchElementException("Array is empty.^")^n return
get(random.nextInt(size))^n}^n/n/**^n * Returns a random element from
this array using the specified source of randomness.^n * ^n * @throws NoSuchElementException if this array is
empty.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun ULongArray.random(random:
Random): ULong {^n if (isEmpty())^n throw NoSuchElementException("Array is empty.^")^n return
get(random.nextInt(size))^n}^n/n/**^n * Returns a random element from this array using the specified source of
randomness.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UByteArray.random(random: Random):
UByte {^n if (isEmpty())^n throw NoSuchElementException("Array is empty.^")^n return
get(random.nextInt(size))^n}^n/n/**^n * Returns a random element from this array using the specified source of
randomness.^n * ^n * @throws NoSuchElementException if this array is empty.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UShortArray.random(random: Random):
UShort {^n if (isEmpty())^n throw NoSuchElementException("Array is empty.^")^n return
get(random.nextInt(size))^n}^n/n/**^n * Returns a random element from this array, or `null` if this array is empty.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@WasExperimental(ExperimentalStdlibApi::class)^n
^@kotlin.internal.InlineOnly^public inline fun UIntArray.randomOrNull(): UInt? {^n return
randomOrNull(Random)^n}^n/n/**^n * Returns a random element from this array, or `null` if this array is empty.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@WasExperimental(ExperimentalStdlibApi::class)^n
^@kotlin.internal.InlineOnly^public inline fun ULongArray.randomOrNull(): ULong? {^n return
randomOrNull(Random)^n}^n/n/**^n * Returns a random element from this array, or `null` if this array is empty.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@WasExperimental(ExperimentalStdlibApi::class)^n
^@kotlin.internal.InlineOnly^public inline fun UByteArray.randomOrNull():
UByte? {^n return randomOrNull(Random)^n}^n/n/**^n * Returns a random element from this array, or `null` if
this array is empty.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@WasExperimental(ExperimentalStdlibApi::class)^n

```

```

@kotlin.internal.InlineOnly\npublic inline fun UShortArray.randomOrNull(): UShort? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npu
blic fun UIntArray.randomOrNull(random: Random): UInt? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npu
blic fun ULongArray.randomOrNull(random:
Random): ULong? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns
a random element from this array using the specified source of randomness, or `null` if this array is empty.\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npu
blic fun UByteArray.randomOrNull(random: Random): UByte? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npu
blic fun UShortArray.randomOrNull(random: Random): UShort? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.single(): UInt {\n return storage.single().toUInt()\n}\n\n/**\n * Returns the single element,
or throws an exception if the array is empty or has more than one element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(): ULong {\n return storage.single().toULong()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(): UByte {\n return storage.single().toUByte()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(): UShort {\n return
storage.single().toUShort()\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws
exception if there is no or more than one matching element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(predicate: (UInt) -> Boolean): UInt {\n var single: UInt? = null\n var found = false\n for
(element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n single = element\n found = true\n }\n }\n if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as UInt\n}\n\n/**\n * Returns the single element matching
the given [predicate], or throws exception if there is no or more than one matching element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.single(predicate: (ULong) -> Boolean): ULong {\n var single: ULong? = null\n var
found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw
IllegalArgumentException("Array contains more than one matching element.")\n single = element\n
found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as ULong\n}\n\n/**\n * Returns the
single element matching the given [predicate], or throws exception if there is no or more than one matching
element.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UByteArray.single(predicate: (UByte) -> Boolean): UByte {\n var single: UByte? = null\n var found =

```

```

false\n for (element in this) {\n if (predicate(element)) {\n
if (found) throw IllegalArgumentException("\nArray contains more than one matching element.\n")\n single =
element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("\nArray contains no
element matching the predicate.\n")\n @Suppress("UNCHECKED_CAST")\n return single as
UByte\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or
more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(predicate: (UShort) -> Boolean): UShort {\n var single: UShort? = null\n var found = false\n
for (element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException("\nArray
contains more than one matching element.\n")\n single = element\n found = true\n }\n }\n if
(!found) throw NoSuchElementException("\nArray contains
no element matching the predicate.\n")\n @Suppress("UNCHECKED_CAST")\n return single as
UShort\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {\n
return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.singleOrNull(): ULong? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull(): UByte? {\n
return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.singleOrNull(): UShort? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n var single: UInt? = null\n var found =
false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n
var single: ULong? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n
if (found) return null\n single = element\n found = true\n }\n }\n if (!found) return null\n
return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not
found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n var single: UByte? = null\n var found =
false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n var single: UShort? = null\n var
found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single
= element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list
containing all elements except first [n] elements.\n *\n * @throws IllegalArgumentException if [n] is negative.\n *\n
*\n@sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n *\n *\n
*\n@throws

```



## IllegalArgumentException

```
if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort>\n{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.dropLast(n: Int): List<UInt> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.dropLast(n: Int):\nList<ULong> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.dropLast(n: Int): List<UByte>\n{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws\nIllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.dropLast(n: Int):\nList<UShort> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size -\n n).coerceAtLeast(0))\n}\n\n**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n**\n * Returns a list containing all elements except last elements that satisfy the\n * given [predicate].\n * \n * @sample\nsamples.collections.Collections.Transformations.drop
```

```

*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n for (index in lastIndex downTo 0)
{\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n for (index in lastIndex downTo
0) {\n if (!predicate(this[index])) {\n return
take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**
 * Returns a list containing all elements except first
elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {\n var yielding = false\n val list =
ArrayList<UInt>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n
 list.add(item)\n yielding = true\n }\n return list\n}\n\n/**
 * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropWhile(predicate: (ULong) -> Boolean):
List<ULong> {\n var yielding = false\n val list = ArrayList<ULong>()\n for (item in this)\n if
(yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n
 }\n return list\n}\n\n/**
 * Returns a list containing all elements except first elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {\n var yielding = false\n val list =
ArrayList<UByte>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item))
{\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**
 * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n
 * \n * @sample samples.collections.Collections.Transformations.drop\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {\n var yielding = false\n val list =
ArrayList<UShort>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item))
{\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**
 * Returns a list containing only
elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {\n return filterTo(ArrayList<UInt>(),
predicate)\n}\n\n/**
 * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {\n return filterTo(ArrayList<ULong>(),
predicate)\n}\n\n/**
 * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {\n return filterTo(ArrayList<UByte>(),
predicate)\n}\n\n/**
 * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {\n return filterTo(ArrayList<UShort>(),

```

predicate)\n}\n\n/\*\*\n

\* Returns a list containing only elements matching the given [predicate].\n \* @param [predicate] function that takes the index of an element and the element itself\n \* and returns the result of predicate evaluation on the element.\n \* \n \* @sample samples.collections.Collections.Filtering.filterIndexed\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {\n return
```

filterIndexedTo(ArrayList<UInt>(), predicate)\n}\n\n/\*\*\n \* Returns a list containing only elements matching the given [predicate].\n \* @param [predicate] function that takes the index of an element and the element itself\n \* and returns the result of predicate evaluation on the element.\n \* \n \* @sample samples.collections.Collections.Filtering.filterIndexed\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.filterIndexed(predicate:
```

```
(index: Int, ULong) -> Boolean): List<ULong> {\n return filterIndexedTo(ArrayList<ULong>(),\n predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n return
```

filterIndexedTo(ArrayList<UByte>(), predicate)\n}\n\n/\*\*\n \* Returns a list containing only elements matching the given [predicate].\n \* @param [predicate] function that takes the index of an element and the element itself\n \* and returns the result of predicate evaluation on the element.\n \* \n \* @sample samples.collections.Collections.Filtering.filterIndexed\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n return\n filterIndexedTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate]\n * to the given [destination].\n * @param [predicate] function that takes the index of an element and the element\n * itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) -> Boolean): C {\n for<each>Indexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int, ULong) -> Boolean): C {\n for<each>Indexed { index, element ->\n if (predicate(index, element))\n destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun <C : MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) -> Boolean): C {\n for<each>Indexed { index, element ->\n if (predicate(index, element))\n destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
```

```

samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) ->
Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element))
destination.add(element)\n }\n return destination\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {\n return filterNotTo(ArrayList<UInt>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {\n return filterNotTo(ArrayList<ULong>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {\n return
filterNotTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {\n return
filterNotTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements not matching the given [predicate]
to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends
all elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterNotTo(destination: C, predicate: (ULong) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends
all elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements
not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends
all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {\n for (element
in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n

```

```

* \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list
containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n *
Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):
List<UShort> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):
List<UInt> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<UInt>(size)\n for (index
in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.slice(indices: Iterable<Int>): List<ULong> {\n val size = indices.collectionSizeOrDefault(10)\n if
(size == 0) return emptyList()\n val list = ArrayList<ULong>(size)\n for (index in indices) {\n
list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: Iterable<Int>):
List<UByte> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<UByte>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.slice(indices: Iterable<Int>): List<UShort> {\n val size = indices.collectionSizeOrDefault(10)\n
if (size == 0) return emptyList()\n val list = ArrayList<UShort>(size)\n for (index in indices) {\n
list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns an array containing elements of this array at specified
[indices].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices:
Collection<Int>): UIntArray {\n return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array
containing elements of this array at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices:

```

```

Collection<Int>: ULongArray {
 return ULongArray(storage.sliceArray(indices))
}

Returns an array containing elements of this array at specified [indices].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
 return
 UByteArray(storage.sliceArray(indices))
}

Returns an array containing elements of this array at specified [indices].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
 return
 UShortArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UIntArray.sliceArray(indices: IntRange): UIntArray {
 return UIntArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 ULongArray.sliceArray(indices: IntRange):
 ULongArray {
 return ULongArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UByteArray.sliceArray(indices: IntRange): UByteArray {
 return
 UByteArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UShortArray.sliceArray(indices: IntRange): UShortArray {
 return
 UShortArray(storage.sliceArray(indices))
}

Returns a list containing first [n] elements.

Throws IllegalArgumentException if [n] is negative.

@sample
samples.collections.Collections.Transformations.take

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UIntArray.take(n: Int): List<UInt> {
 require(n >= 0) { "Requested element count $n is less than zero." }
 if (n == 0) return emptyList()
 if (n >= size) return toList()
 if (n == 1) return listOf(this[0])
 var count = 0
 val list = ArrayList<UInt>(n)
 for (item in this) {
 list.add(item)
 if (++count == n) break
 }
 return list
}

Returns a list containing first [n] elements.

Throws IllegalArgumentException if [n] is negative.

@sample
samples.collections.Collections.Transformations.take

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 ULongArray.take(n: Int): List<ULong> {
 require(n >= 0) { "Requested element count $n is less than zero." }
 if (n == 0) return emptyList()
 if (n >= size) return toList()
 if (n == 1) return listOf(this[0])
 var count = 0
 val list = ArrayList<ULong>(n)
 for (item in this) {
 list.add(item)
 if (++count == n) break
 }
 return list
}

Returns a list containing first [n] elements.

Throws IllegalArgumentException if [n] is negative.

@sample
samples.collections.Collections.Transformations.take

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UByteArray.take(n: Int): List<UByte> {
 require(n >= 0) { "Requested element count $n is less than zero." }
 if (n == 0) return emptyList()
 if (n >= size) return toList()
 if (n == 1) return listOf(this[0])
 var count = 0
 val list = ArrayList<UByte>(n)
 for (item in this) {
 list.add(item)
 if (++count == n) break
 }
 return list
}

Returns a list containing first [n] elements.

Throws IllegalArgumentException if [n] is negative.

@sample
samples.collections.Collections.Transformations.take

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UShortArray.take(n: Int): List<UShort> {
 require(n >= 0) { "Requested element count $n is less than zero." }
 if (n == 0) return emptyList()
 if (n >= size) return toList()
 if (n == 1) return listOf(this[0])
 var count = 0
 val list = ArrayList<UShort>(n)
 for (item in this) {
 list.add(item)
 if (++count == n) break
 }
 return list
}

Returns a list containing last [n] elements.

Throws IllegalArgumentException if [n] is negative.

@sample
samples.collections.Collections.Transformations.take

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
 UIntArray.takeLast(n: Int): List<UInt> {

```

```

require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<UInt>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.takeLast(n: Int): List<ULong> {\n require(n >= 0) { \"Requested element count $n is less than
zero.\" } \n if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return
listOf(this[size - 1])\n val list = ArrayList<ULong>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentExcep\n
tion if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.takeLast(n: Int): List<UByte>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UByte>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentExcep\n
tion if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.takeLast(n: Int): List<UShort>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" } \n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UShort>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt>
{\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n for (index in lastIndex downTo 0)
{\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n for (index
in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n for (index in lastIndex downTo 0)
{\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {\n val list = ArrayList<UInt>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample

```

```

samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {\n val list = ArrayList<ULong>()\n for
(item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {\n val list = ArrayList<UByte>()\n for
(item in this)
{\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list
containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {\n val list = ArrayList<UShort>()\n for
(item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.reverse(fromIndex: Int, toIndex:
Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex,
toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex
the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *

```



Returns a list with elements in reversed order.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.reversed(): List<UInt> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.reversed(): List<ULong> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.reversed(): List<UByte> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.reversed(): List<UShort> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reversedArray(): UIntArray {\n return UIntArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reversedArray(): ULongArray {\n return ULongArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reversedArray(): UByteArray {\n return UByteArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reversedArray(): UShortArray {\n return UShortArray(storage.reversedArray())\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See:
```

[https://en.wikipedia.org/wiki/Fisher%20%93Yates\\_shuffle#The\\_modern\\_algorithm](https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm)

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See:
```

[https://en.wikipedia.org/wiki/Fisher%20%93Yates\\_shuffle#The\\_modern\\_algorithm](https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm)

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See:
```

[https://en.wikipedia.org/wiki/Fisher%20%93Yates\\_shuffle#The\\_modern\\_algorithm](https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm)

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See:
```

[https://en.wikipedia.org/wiki/Fisher%20%93Yates\\_shuffle#The\\_modern\\_algorithm](https://en.wikipedia.org/wiki/Fisher%20%93Yates_shuffle#The_modern_algorithm)

```

*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^public fun UShortArray.shuffle(random: Random):
Unit {^n for (i in lastIndex downTo 1) {^n val j = random.nextInt(i + 1)^n val copy = this[i]^n this[i] =
this[j]^n this[j] = copy^n }^n}^n/^n/^n * Sorts elements in the array in-place descending according to their
natural sort order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public
fun UIntArray.sortDescending(): Unit {^n if (size > 1) {^n sort()^n reverse()^n }^n}^n/^n/^n * Sorts
elements in the array in-place descending according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun ULongArray.sortDescending(): Unit {^n if
(size > 1) {^n sort()^n reverse()^n }^n}^n/^n/^n * Sorts elements in the array in-place descending
according to their natural sort order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun
UByteArray.sortDescending(): Unit {^n if (size > 1) {^n sort()^n reverse()^n }^n}^n/^n/^n * Sorts
elements in the array in-place descending according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UShortArray.sortDescending(): Unit {^n
if (size > 1) {^n sort()^n reverse()^n }^n}^n/^n/^n * Returns a list of all elements sorted according to their
natural sort
order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UIntArray.sorted(): List<UInt> {^n
return copyOf().apply { sort() }.asList()^n}^n/^n/^n * Returns a list of all elements sorted according to their natural
sort order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun ULongArray.sorted():
List<ULong> {^n return copyOf().apply { sort() }.asList()^n}^n/^n/^n * Returns a list of all elements sorted
according to their natural sort order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun
UByteArray.sorted(): List<UByte> {^n return copyOf().apply { sort() }.asList()^n}^n/^n/^n * Returns a list of all
elements sorted according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UShortArray.sorted(): List<UShort> {^n
return copyOf().apply { sort() }.asList()^n}^n/^n/^n * Returns an array with all elements of this array sorted
according to their natural sort order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public
fun UIntArray.sortedArray(): UIntArray {^n if (isEmpty()) return this^n return this.copyOf().apply { sort()
}^n}^n/^n/^n * Returns an array with all elements of this array sorted according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun ULongArray.sortedArray(): ULongArray
{^n if (isEmpty()) return this^n return this.copyOf().apply { sort() }^n}^n/^n/^n * Returns an array with all
elements of this array sorted according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UByteArray.sortedArray(): UByteArray {^n
if (isEmpty()) return this^n return this.copyOf().apply { sort() }^n}^n/^n/^n * Returns an array with all elements
of this array sorted according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UShortArray.sortedArray(): UShortArray
{^n if (isEmpty()) return this^n
return this.copyOf().apply { sort() }^n}^n/^n/^n * Returns an array with all elements of this array sorted
descending according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UIntArray.sortedArrayDescending():
UIntArray {^n if (isEmpty()) return this^n return this.copyOf().apply { sortDescending() }^n}^n/^n/^n * Returns
an array with all elements of this array sorted descending according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun ULongArray.sortedArrayDescending():
ULongArray {^n if (isEmpty()) return this^n return this.copyOf().apply { sortDescending() }^n}^n/^n/^n *
Returns an array with all elements of this array sorted descending according to their natural sort order.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun UByteArray.sortedArrayDescending():
UByteArray {^n if (isEmpty()) return this^n return this.copyOf().apply { sortDescending()
}^n}^n/^n/^n * Returns an array with all elements of this array sorted descending according to their natural sort
order.^n *^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^public fun
UShortArray.sortedArrayDescending(): UShortArray {^n if (isEmpty()) return this^n return this.copyOf().apply {
sortDescending() }^n}^n/^n/^n * Returns a list of all elements sorted descending according to their natural sort

```

order.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.sortedDescending(): List<UInt> {\n return copyOf().apply { sort() }.reversed()\n }\n\n \*\*\n \* Returns a list of all elements sorted descending according to their natural sort order.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.sortedDescending(): List<ULong> {\n return copyOf().apply { sort() }.reversed()\n }\n\n \*\*\n \* Returns a list of all elements sorted descending according to their natural sort order.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UByteArray.sortedDescending(): List<UByte> {\n return copyOf().apply { sort() }.reversed()\n }\n\n \*\*\n \* Returns a list of all elements sorted descending according to their natural sort order.\n \* \n \* The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UShortArray.sortedDescending(): List<UShort> {\n return copyOf().apply { sort() }.reversed()\n }\n\n \*\*\n \* Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UByteArray.asByteArray(): ByteArray {\n return storage\n }\n\n \*\*\n \* Returns an array of type [IntArray], which is a view of this array where each element is a signed reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UIntArray.asIntArray(): IntArray {\n return storage\n }\n\n \*\*\n \* Returns a [List] that wraps the original array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public expect fun UIntArray.asList(): List<UInt>\n\n \*\*\n \* Returns a [List] that wraps the original array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public expect fun ULongArray.asList(): List<ULong>\n\n \*\*\n \* Returns a [List] that wraps the original array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public expect fun UByteArray.asList(): List<UByte>\n\n \*\*\n \* Returns a [List] that wraps the original array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n public expect fun UShortArray.asList(): List<UShort>\n\n \*\*\n \* Returns an array of type [LongArray], which is a view of this array where each element is a signed reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun ULongArray.asLongArray(): LongArray {\n return storage\n }\n\n \*\*\n \* Returns an array of type [ShortArray], which is a view of this array where each element is a signed reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UShortArray.asShortArray(): ShortArray {\n return storage\n }\n\n \*\*\n \* Returns an array of type [UByteArray], which is a view of this array where each element is an unsigned reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun ByteArray.asUByteArray(): UByteArray {\n return UByteArray(this)\n }\n\n \*\*\n \* Returns an array of type [UIntArray], which is a view of this array where each element is an unsigned reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun IntArray.asUIntArray(): UIntArray {\n return UIntArray(this)\n }\n\n \*\*\n \* Returns an array of type [ULongArray], which is a view of this array where each element is an unsigned reinterpretation\n \* of the corresponding element of this array.\n \* \n \* Since Kotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun LongArray.asULongArray(): ULongArray {\n return ULongArray(this)\n }\n\n \*\*\n \* Returns an array of type

[UShortArray],

which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun ShortArray.asUShortArray(): UShortArray {\n return UShortArray(this)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray.contentEquals(other: UIntArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray.contentEquals(other: ULongArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray.contentEquals(other: UByteArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray.contentEquals(other: UShortArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin("1.4")@ExperimentalUnsignedTypes\npublic infix fun UIntArray?.contentEquals(other: UIntArray?): Boolean {\n return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin("1.4")@ExperimentalUnsignedTypes\npublic infix fun ULongArray?.contentEquals(other: ULongArray?): Boolean {\n return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin("1.4")@ExperimentalUnsignedTypes\npublic infix fun UByteArray?.contentEquals(other: UByteArray?): Boolean {\n return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin("1.4")@ExperimentalUnsignedTypes\npublic infix fun UShortArray?.contentEquals(other: UShortArray?): Boolean {\n return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.3")@DeprecatedSinceKotlin(hiddenSince = "1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentHashCode(): Int {\n return this.contentHashCode()\n}
```



```

return this?.joinToString(", ", "[", "]") ?: "null"
}

Returns a string representation of the contents of
the specified array as if it is [List].

@sample
samples.collections.Arrays.ContentOperations.contentToString

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
public
fun UShortArray?.contentToString(): String {
 return this?.joinToString(", ", "[", "]") ?: "null"
}

Copies this array or its subrange into the [destination] array and returns that array.
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
@param destination the array to copy to.
@param destinationOffset the position in the [destination] array to copy to, 0 by default.
@param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
@throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
@return the [destination] array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
UIntArray {
 storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
 return destination
}

Copies this array or its subrange into the [destination] array and returns that array.
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
@param destination the array to copy to.
@param destinationOffset the position in the [destination] array to copy to, 0 by default.
@param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
@throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
@return the [destination] array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
ULongArray {
 storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
 return destination
}

Copies this array or its subrange into the [destination] array and returns that array.
It's allowed to pass the same array in the [destination] and even specify the
subrange so that it overlaps with the destination range.
@param destination the array to copy to.
@param destinationOffset the position in the [destination] array to copy to, 0 by default.
@param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
@throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
@return the [destination] array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex:
Int = 0, endIndex: Int = size): UByteArray {
 storage.copyInto(destination.storage, destinationOffset, startIndex,
endIndex)
 return destination
}

Copies this array or its subrange into the [destination] array and
returns that array.
It's allowed to pass the same array in the [destination] and even specify the subrange so
that it overlaps with the destination range.
@param destination the array to copy to.
@param destinationOffset the position in the [destination] array to copy to, 0 by default.
@param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
@param endIndex the end (exclusive) of the

```

subrange to copy, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \* or when that index is out of the [destination] array indices range.\n \* \n \* @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =\nsize): UShortArray {\n storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n return\n destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample\n samples.collections.Arrays.CopyOfOperations.copyOfOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.copyOf(): UIntArray {\n return UIntArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is\n a copy of the original array.\n * \n * @sample\n samples.collections.Arrays.CopyOfOperations.copyOfOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun\nULongArray.copyOf(): ULongArray {\n return ULongArray(storage.copyOf())\n}\n\n/**\n * Returns\n new array which is a copy of the original array.\n * \n * @sample\n samples.collections.Arrays.CopyOfOperations.copyOfOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.copyOf(): UByteArray {\n return UByteArray(storage.copyOf())\n}\n\n/**\n * Returns new array\n which is a copy of the original array.\n * \n * @sample\n samples.collections.Arrays.CopyOfOperations.copyOfOf\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.copyOfOf(): UShortArray {\n return UShortArray(storage.copyOf())\n}\n\n/**\n * Returns new array\n which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at\n the end with zero values if necessary.\n * \n * - If [newSize] is
```

less than the size of the original array, the copy array is truncated to the [newSize].\n \* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.copyOfOf(newSize: Int): UIntArray {\n return UIntArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either\n truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the\n original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original\n array, the extra elements in the copy array are filled with zero values.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.copyOfOf(newSize: Int): ULongArray\n\n{\n return ULongArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original\n array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if\n necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the\n [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are\n filled with zero values.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.copyOfOf(newSize: Int): UByteArray {\n return UByteArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either\n truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the\n original array, the copy array is truncated to the [newSize].\n * - If [newSize]\n is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.copyOfOf(newSize: Int): UShortArray {\n return UShortArray(storage.copyOfOf(newSize))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start
```

of the range (inclusive) to copy.\n \* @param toIndex the end of the range (exclusive) to copy.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {\n return
UIntArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
 * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the
 start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n *
 @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
 array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
 ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {\n return
 ULongArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
 * Returns a new array which is a copy of the
 specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
 @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
 [fromIndex] is less than zero or [toIndex] is greater than the size of
 this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
 UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {\n return
 UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
 * Returns a new array which is a copy of the
 specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
 @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
 [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
 UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {\n return
 UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the
 range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this
 array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
 greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.fill(element: UInt, fromIndex:
 Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toInt(), fromIndex, toIndex)\n}\n\n/**\n
 * Fills this array
 or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive)
 to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
 default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than
 zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
 greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
 ULongArray.fill(element: ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toLong(),
 fromIndex, toIndex)\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n *
 @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the
 range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex]
 is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex]
 is greater than [toIndex].\n
 *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
 UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toByte(),
 fromIndex,
 toIndex)\n}\n\n/**\n
 * Fills this array or its subrange with the specified [element] value.\n * \n * @param
 fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
 to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than
 zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
 greater than [toIndex].\n
 */

```



to fill, size of this array by default.\n \* \n \* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n \* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n \*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toShort(), fromIndex, toIndex)\n}\n\n\*\*\n \* Returns the range of valid indices for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.indices: IntRange\n get() = storage.indices\n\n\*\*\n \* Returns the range of valid indices for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.indices: IntRange\n get() = storage.indices\n\n\*\*\n \* Returns the range of valid indices for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UShortArray.indices: IntRange\n get() = storage.indices\n\n\*\*\n \* Returns the last valid index for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.lastIndex: Int\n get() = storage.lastIndex\n\n\*\*\n \* Returns the last valid index for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.lastIndex: Int\n get() = storage.lastIndex\n\n\*\*\n \* Returns the last valid index for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UByteArray.lastIndex: Int\n get() = storage.lastIndex\n\n\*\*\n \* Returns the last valid index for the array.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UShortArray.lastIndex: Int\n get() = storage.lastIndex\n\n\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.plus(element: UInt): UIntArray {\n return UIntArray(storage + element.toInt())\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.plus(element: ULong): ULongArray {\n return ULongArray(storage + element.toLong())\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.plus(element: UByte): UByteArray {\n return UByteArray(storage + element.toByte())\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then the given [element].\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.plus(element: UShort): UShortArray {\n return UShortArray(storage + element.toShort())\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic operator fun UIntArray.plus(elements: Collection<UInt>): UIntArray {\n var index = size\n val result = storage.copyOf(size + elements.size)\n for (element in elements) result[index++] = element.toInt()\n return UIntArray(result)\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic operator fun ULongArray.plus(elements: Collection<ULong>): ULongArray {\n var index = size\n val result = storage.copyOf(size + elements.size)\n for (element in elements) result[index++] = element.toLong()\n return ULongArray(result)\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {\n var index = size\n val result = storage.copyOf(size + elements.size)\n for (element in elements) result[index++] = element.toByte()\n return UByteArray(result)\n}\n\n\*\*\n \* Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n\n\*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic

```

operator fun UShortArray.plus(elements: Collection<UShort>): UShortArray {\n var index = size\n val result =
storage.copyOfOf(size + elements.size)\n for (element in elements) result[index++] = element.toShort()\n return
UShortArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] array.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UIntArray.plus(elements: UIntArray): UIntArray {\n return UIntArray(storage + elements.storage)\n}\n\n/**\n *
Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
ULongArray.plus(elements: ULongArray): ULongArray {\n return ULongArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing all elements
of the original array and then all elements of the given [elements] array.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UByteArray.plus(elements: UByteArray): UByteArray {\n return UByteArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] array.\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UShortArray.plus(elements: UShortArray): UShortArray {\n return UShortArray(storage +
elements.storage)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.sort(): Unit {\n if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n *
@sample samples.collections.Arrays.Sorting.sortArray\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.sort(): Unit {\n if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n *
@sample samples.collections.Arrays.Sorting.sortArray\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(): Unit {\n if (size > 1)
sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.sort(): Unit {\n if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts a range in the array in-
place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end
of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentOutOfRangeException if
[fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(fromIndex: Int = 0, toIndex:
Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this, fromIndex,
toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentOutOfRangeException if [fromIndex] is greater than [toIndex].\n *
*\n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n sortArray(this, fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n *
@param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range
(exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentOutOfRangeException if
[fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(fromIndex: Int = 0,

```

```

toIndex: Int = size): Unit { \n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive)
to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit { \n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are
sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive)
to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentExcepion
if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit { \n sort(fromIndex, toIndex)\n reverse(fromIndex,
toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted
descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to
sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentExcepion if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit { \n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements
of the array in the specified range
in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex
the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit { \n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements
of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort
order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range
(exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit { \n sort(fromIndex, toIndex)\n reverse(fromIndex,
toIndex)\n}\n\n/**\n * Returns an array of type [ByteArray], which is a copy of this array where
each element is a signed reinterpretation\n * of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.toByteArray(): ByteArray { \n return storage.copyOf()\n}\n\n/**\n * Returns an array of type
[IntArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding
element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.toIntArray(): IntArray { \n return storage.copyOf()\n}\n\n/**\n * Returns an array of type [LongArray], which is a copy of this array where each element is a signed
reinterpretation\n * of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.toLongArray(): LongArray { \n return storage.copyOf()\n}\n\n/**\n * Returns an array of type
[ShortArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding

```

element of this array.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.toShortArray(): ShortArray {\n return storage.copyOfOf()\n}\n\n/**\n * Returns a *typed* object array\n containing all of the elements of this primitive array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt>\n{\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object\n array containing all of the elements of this primitive array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.toTypedArray():\nArray<ULong> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array\n containing all of the elements of this primitive array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray():\nArray<UByte> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array\n containing all of the elements of this primitive array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray():\nArray<UShort> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of UByte containing\n all of the elements of this generic array.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nArray<out UByte>.toUByteArray(): UByteArray {\n return UByteArray(size) { index -> this[index]\n }\n}\n\n/**\n * Returns an array of type [UByteArray], which is a copy of this array where each element is an\n unsigned reinterpretation\n * of the corresponding element of this array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nByteArray.toUByteArray(): UByteArray {\n return UByteArray(this.copyOf())\n}\n\n/**\n * Returns an array of\n UInt containing all of the elements of this generic array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UInt>.toUIntArray(): UIntArray\n{\n return UIntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UIntArray], which is a\n copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this\n array.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nIntArray.toUIntArray(): UIntArray {\n return UIntArray(this.copyOf())\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this generic array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nIntArray.toUIntArray(): UIntArray {\n return UIntArray(this.copyOf())\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this generic array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out ULong>.toULongArray():\nULongArray {\n return ULongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type\n [ULongArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the\n corresponding element of this array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nLongArray.toULongArray(): ULongArray {\n return ULongArray(this.copyOf())\n}\n\n/**\n * Returns an array\n of UShort containing all of the elements of this generic array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UShort>.toUShortArray():\nUShortArray {\n return UShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type\n [UShortArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the\n corresponding element of this array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nShortArray.toUShortArray(): UShortArray {\n return UShortArray(this.copyOf())\n}\n\n/**\n * Returns a [Map]\n where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to\n each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map\n preserves the entry iteration order of the original array.\n * \n * @sample\n samples.collections.Collections.Transformations.associateWith\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>\nUIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {\n val result = LinkedHashMap<UInt,\n V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a
```

[Map] where keys are elements

from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Collections.Transformations.associateWith

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
```

```
ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {\n val result =\n LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result,
```

```
valueSelector)\n}\n\n* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.
```

@sample

samples.collections.Collections.Transformations.associateWith

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
```

```
UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {\n val result =\n LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result,
```

```
valueSelector)\n}\n\n* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.
```

@sample samples.collections.Collections.Transformations.associateWith

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
```

```
UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n val result =\n LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n* Populates and returns the [destination] mutable map
```

```
with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
```

@sample samples.collections.Collections.Transformations.associateWithTo

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M >
```

```
MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n* Populates and returns the [destination] mutable map with key-value pairs for each element
```

```
of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
```

@sample samples.collections.Collections.Transformations.associateWithTo

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M >
```

```
MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return
```

```
destination\n}\n\n* Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
```

@sample samples.collections.Collections.Transformations.associateWithTo

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
```

```
inline fun <V, M : MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n* Populates and returns the [destination] mutable map with key-value pairs for each
```

```
element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
```

@sample samples.collections.Collections.Transformations.associateWithTo

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M
{\n for (element in this) {\n destination.put(element,
valueSelector(element))\n }\n return destination\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded
from results of [transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n
 * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
 * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
 * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun <R> UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) ->
Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and
its index in the original array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapIndexedTo(destination: C, transform: (index: Int, ULong) ->
Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original
array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapIndexedTo(destination:
C, transform: (index: Int, UByte) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list =
transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element\n * and its index in the original
array, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) ->
Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked
on each element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being
invoked on each element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {\n for
(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :

```

```

MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {
 for (element in this) {
 val list = transform(element)
 destination.addAll(list)
 }
 return destination
}

/**
 * Groups elements of the original array by the key returned by
 * the given [keySelector] function
 * applied to each element and returns a map where each group key is associated
 * with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys
 * produced from the original array.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K>
UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {
 return groupByTo(LinkedHashMap<K,
 MutableList<UInt>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the
 * given [keySelector] function
 * applied to each element and returns a map where each group key is associated with
 * a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced
 * from the original array.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun <K> ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {
 return
 groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)
}

/**
 * Groups elements of the
 * original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map
 * where each group key is associated with a list of corresponding elements.
 * The returned map preserves the
 * entry iteration order of the keys produced from the original array.
 * @sample
 * samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K>
UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {
 return
 groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)
}

/**
 * Groups elements of the
 * original array by the key returned by the given [keySelector]
 * function
 * applied to each element and returns a map where each group key is associated with a list of
 * corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the
 * original array.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K>
UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {
 return
 groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)
}

/**
 * Groups values returned by
 * the [valueTransform] function applied to each element of the original array
 * by the key returned by the given
 * [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of
 * corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the
 * original array.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V>
UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {
 return
 groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups values
 * returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by
 * the given [keySelector] function applied to the element
 * and returns a map where each group key is associated
 * with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys
 * produced from the original array.
 * @sample
 * samples.collections.Collections.Transformations.groupByKeysAndValues
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V>
ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform:
(ULong) -> V): Map<K, List<V>> {
 return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
 valueTransform)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of
 * the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns
 * a map where each group key is associated with a list of corresponding values.
 * The returned map preserves
 * the entry iteration order of the keys produced from the original array.
 * @sample
 */

```



```

samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element
of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and
returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map
preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements
of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to
the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {\n
for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<UInt>() }\n list.add(element)\n }\n return destination}\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong)
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<ULong>() }\n list.add(element)\n }\n return destination}\n}\n\n/**\n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K):
M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<UByte>() }\n list.add(element)\n }\n return destination}\n}\n\n/**\n * Groups elements
of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to
the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K):
M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<UShort>() }\n list.add(element)\n }\n return destination}\n}\n\n/**\n * Groups values returned by
the [valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and puts to the [destination] map each group key associated
with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K,
valueTransform: (UInt) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return

```

```

destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K,
valueTransform: (ULong) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list
= destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector:
(UByte) -> K, valueTransform: (UByte) -> V): M {\n for (element in this) {\n val key =
keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n
list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a
list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UShortArray.groupByTo(destination: M, keySelector:
(UShort) -> K, valueTransform: (UShort) -> V): M {\n for (element in this) {\n val key =
keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n
list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Returns a list containing the results of
applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.map(transform: (UInt) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to each element in the original
array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.map(transform:
(ULong) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.map(transform: (UByte) -> R): List<R> {\n return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.map(transform: (UShort) -> R): List<R> {\n return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given
[transform] function\n * to each element and its index in the original array.\n * @param [transform] function that
takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UIntArray.mapIndexed(transform: (index: Int, UInt) -> R): List<R> {\n return

```

```

mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> ULongArray.mapIndexed(transform: (index: Int, ULong) -> R): List<R> {\n return
mapIndexedTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element and its index in the original array.\n * @param [transform] function that takes the index of an element and
the element itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.mapIndexed(transform: (index: Int, UByte) -> R): List<R> {\n return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UShortArray.mapIndexed(transform:
(index: Int, UShort) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n *
Applies the given [transform] function to each element and its index in the original array\n * and appends the results
to the given [destination].\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {\n var
index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n *
Applies the given [transform] function to each element and its index in the original array\n * and appends the results
to the given [destination].\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {\n var
index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n
* and appends the results to the given [destination].\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {\n var
index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n *
Applies the given [transform] function to each element and its index in the original array\n * and appends the results
to the given [destination].\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {\n var
index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and
appends the results to the given [destination].\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {\n for (item in this)\n
destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each

```

element of the original array\n \* and appends the results to the given [destination].\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {\n for (item in
this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {\n for (item in
this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps
each element of the original array\n * into an [IndexedValue] containing the index of that element and the element
itself.\n *@\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.withIndex():
```

```
Iterable<IndexedValue<UInt>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n *@\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {\n return IndexingIterable { iterator()
}\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue]
containing the index of that element and the element itself.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.withIndex():
Iterable<IndexedValue<UByte>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n *@\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.withIndex():
Iterable<IndexedValue<UShort>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns `true` if all
elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns
`true`\n
```

\* because there are no elements in it that `_do not_` match the predicate.\n \* See a more detailed explanation of this  
logic concept in [`Vacuous truth`]([https://en.wikipedia.org/wiki/Vacuous\\_truth](https://en.wikipedia.org/wiki/Vacuous_truth)) article.\n \* \n \* @sample  
samples.collections.Collections.Aggregates.all\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.all(predicate: (UInt) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the
array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the
predicate.\n * See a more detailed explanation of this logic concept in [Vacuous
truth](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.all(predicate: (ULong) -> Boolean): Boolean {\n for (element in this) if
(!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given
[predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no
elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in
[Vacuous truth](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the
```

array contains no elements, the function

returns `true` because there are no elements in it that `_do not_` match the predicate. See a more detailed explanation of this logic concept in ["Vacuous truth"](https://en.wikipedia.org/wiki/Vacuous\_truth) article.

@sample samples.collections.Collections.Aggregates.all

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.any(): Boolean {\n return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.any(): Boolean {\n return storage.any()\n}\n\n/**\n * Returns `true` if array has at least
one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.any(): Boolean {\n return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one
element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\n return storage.any()\n}\n\n/**\n * Returns `true` if at least one element matches
the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.any(predicate: (UInt) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one
element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.any(predicate: (ULong) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.any(predicate: (UByte) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
true\n return false\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.count(predicate: (UInt) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements
matching the given [predicate].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.count(predicate: (UByte) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {\n var count = 0\n for (element in this) if
```

(predicate(element)) ++count\n return count\n}\n\n/\*\*\n \* Accumulates value starting with [initial] value and applying [operation] from left to right\n \* to current accumulator value and each element.\n \* \n \* Returns the specified [initial] value if the array is empty.\n \* \n \* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {\n var accumulator = initial\n for
(element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and
each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {\n var accumulator = initial\n for (element in
this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value
and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator
value and an element, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {\n var accumulator = initial\n for (element in
this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each
element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {\n var accumulator = initial\n
for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator
value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the
array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n *
and the element itself, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {\n var index = 0\n var
accumulator = initial\n for (element in this) accumulator = operation(index++,
accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {\n var index = 0\n var
accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index
in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {\n var index = 0\n var

```



[operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\n ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {\n var index =\n lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),\n accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]\n * value and applying [operation]
```

from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\n UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {\n var index =\n lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),\n accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]\n * value and applying [operation] from right to left\n * to each element with its index in the original array and current\n * accumulator value.\n * Returns the specified [initial] value if the array is empty.
```

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\n UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {\n var index =\n lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),\n accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Performs the given [action] on each\n * element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline\nfun UIntArray.forEach(action: (UInt) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n ULongArray.forEach(action:\n (ULong) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each\n * element.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline\nfun UByteArray.forEach(action: (UByte) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n UShortArray.forEach(action: (UShort) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action]\n * function that takes the index of an element and the element itself\n * and performs the action on the element.
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit):\n Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on\n * each element, providing sequential index with the element.\n * @param [action] function that takes the index of an\n * element and the element itself\n * and performs the action on the element.
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {\n var index = 0\n for (item in this)\n action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with\n * the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs\n * the action on the element.
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {\n var index = 0\n for (item in
```



```

this) action(index++, item)}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and
performs the action on the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {\n var index = 0\n for (item in this)
action(index++, item)\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the
array is empty.\n *\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.max(): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest
element.\n
* \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.max(): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i
in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the
largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.max(): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic
fun UShortArray.max(): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n
for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns
the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic inline fun <R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt {\n if (isEmpty())
throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return maxElem\n var maxValue = selector(maxElem)\n
 for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n
 maxElem
= e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the
largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n *
@sample samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic inline fun <R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n
 maxElem
= e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element
yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n
* \n * @sample samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")

```

```

\)\npublic inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest
value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*/\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS")\n\)\npublic inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {\n if (isEmpty()) return null\n var
maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue =
selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v)
{\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {\n if (isEmpty()) return null\n var
maxElem = this[0]\n
 val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n
for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem =
e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the
largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {\n if (isEmpty()) return null\n var
maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue =
selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v)
{\n maxElem
= e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the
largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {\n if (isEmpty()) return null\n var
maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue =
selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v)
{\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UIntArray.maxOf(selector: (UInt) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var maxValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun ULongArray.maxOf(selector: (ULong) -> Double): Double {
 if (isEmpty()) throw
 NoSuchElementException()
 var maxValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v =
 selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the
 * largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned
 * result is `NaN`.
 * @throws
 * NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun
UByteArray.maxOf(selector: (UByte) -> Double): Double {
 if (isEmpty()) throw NoSuchElementException()
 var maxValue =
 selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned
 * result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun
UShortArray.maxOf(selector: (UShort) -> Double): Double {
 if (isEmpty()) throw
 NoSuchElementException()
 var maxValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v =
 selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the
 * largest value among all values produced by [selector]
 * function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`,
 * the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun
UIntArray.maxOf(selector: (UInt) -> Float): Float {
 if (isEmpty()) throw NoSuchElementException()
 var
 maxValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 maxValue =
 maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun
ULongArray.maxOf(selector: (ULong) -> Float): Float {
 if (isEmpty()) throw
 NoSuchElementException()
 var maxValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v =
 selector(this[i])
 maxValue = maxOf(maxValue, v)
 }
 return maxValue
}

/**
 * Returns the
 * largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws
 * NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun
UByteArray.maxOf(selector: (UByte)

```

```

-> Float): Float { \n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOf(selector: (UShort) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n
 return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.maxOf(selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> ULongArray.maxOf(selector: (ULong) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxOf(selector: (UByte) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n
 for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n
 }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.maxOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
 }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*/

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.maxOrNull(selector: (ULong) -> Double): Double? {\n if (isEmpty()) return null\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOrNull(selector: (UByte) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
*\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOrNull(selector: (UShort) -> Double): Double? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOrNull(selector: (UInt) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.maxOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UIntArray.maxOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no
elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are
no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UShortArray.maxOrNull(selector: (UShort) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n/**\n * Returns the largest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n
}\n }\n return maxValue\n}\n/**\n * Returns the largest value according to the provided [comparator]\n *
among all values produced by [selector] function applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =

```

```

selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n
* @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n if
(isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
 if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
 if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
 if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxOrNull(): UInt? {\n if
(isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max
= e\n }\n return max\n}\n\n/**\n * Returns
the largest element or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxOrNull(): ULong? {\n if
(isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max
= e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxOrNull(): UByte? {\n if
(isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max
= e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxOrNull(): UShort? {\n if
(isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max
= e\n }\n return
max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n
*\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.maxWith(comparator: Comparator<in UInt>): UInt {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.maxWith(comparator: Comparator<in ULong>): ULong {\n if (isEmpty()) throw
NoSuchElementException()\n
var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max
= e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided
[comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.maxWith(comparator: Comparator<in UByte>): UByte {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic
fun UShortArray.maxWith(comparator: Comparator<in UShort>): UShort {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWithOrNull(comparator:
Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.maxWithOrNull(comparator: Comparator<in ULong>): ULong? {\n if (isEmpty()) return null\n
var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max =
e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided
[comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxWithOrNull(comparator:
Comparator<in UShort>):
UShort? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n
if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n * \n
*\n * @throws NoSuchElementException if the array is empty.\n

```



```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.min(): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.min(): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var
min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return
min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.min(): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UShortArray.min(): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n
 if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element yielding the smallest value of
the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic inline fun <R : Comparable<R>> UIntArray.minBy(selector: (UInt) -> R): UInt {\n if (isEmpty())
throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val
v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic inline fun <R : Comparable<R>> ULongArray.minBy(selector: (ULong) -> R): ULong {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n
 }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic inline fun <R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n
 }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow-
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS
")\npublic
inline fun <R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort {\n if (isEmpty()) throw
NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)
return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v =
selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there
are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minByOrNull(selector: (UInt) -> R): UInt? {\n if (isEmpty()) return null\n var
minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minByOrNull(selector: (ULong) -> R): ULong? {\n if (isEmpty()) return null\n
var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue =
selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v)
{\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minByOrNull(selector: (UByte) -> R): UByte? {\n if (isEmpty()) return null\n
var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue =
selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v)
{\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UShortArray.minByOrNull(selector: (UShort) -> R): UShort? {\n if (isEmpty())
return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n
var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if
(minValue > v) {\n minElem = e\n minValue = v\n }\n }\n
return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.minOf(selector: (UInt) -> Double): Double {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n minValue = minOf(minValue, v)\n }\n
return minValue\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any
of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOf(selector: (ULong) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val
v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOf(selector: (UByte) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOf(selector: (UShort) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.minOf(selector: (UInt) -> Float): Float {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOf(selector: (ULong) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOf(selector: (UByte) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOf(selector: (UShort) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n

```

```

var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n
 for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element
in the array.\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * @throws NoSuchElementException if any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {\n if (isEmpty()) return null\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
 minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {\n if (isEmpty()) return null\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {\n if (isEmpty()) return null\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.minOfOrNull(selector: (UInt) -> Float): Float? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result
is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *

```

applied to each element in the array or `null` if there are no elements.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var\n minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun <R : Comparable<R>> UByteArray.minOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return\n null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if\n (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value\n * among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no\n * elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var\n minValue\n = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided\n * [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw\n NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return\n minValue\n}\n\n/**\n * Returns the\n * smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied\n * to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n if (isEmpty()) throw\n NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return\n minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n if (isEmpty()) throw\n NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =\n selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return\n minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```

<R> UShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {
 if (isEmpty())
 throw NoSuchElementException()
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {
 if (isEmpty()) return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
ULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {
 if (isEmpty()) return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {
 if (isEmpty()) return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {
 if (isEmpty()) return null
 var minValue = selector(this[0])
 for (i in 1..lastIndex) {
 val v = selector(this[i])
 if (comparator.compare(minValue, v) > 0) {
 minValue = v
 }
 }
 return minValue
}

Returns the smallest element or `null` if there are no elements.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.minOrNull(): UInt? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (min > e) min = e
 }
 return min
}

Returns the smallest element or `null` if there are no elements.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.minOrNull(): ULong? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (min > e) min = e
 }
 return min
}

Returns the smallest element or `null` if there are no elements.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.minOrNull(): UByte? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (min > e) min = e
 }
 return min
}

Returns the smallest element or `null` if there are no elements.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.minOrNull(): UShort? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (min > e) min = e
 }
 return min
}

Returns the first element having the smallest value according to the provided [comparator].

@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.7")
@kotlin.jvm.JvmName("minWithOrThrow")
@ExperimentalUnsignedTypes
@Suppress("CONFLICTING_OVERLOADS")
public

```

```

fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt {
 if (isEmpty()) throw NoSuchElementException()
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator].
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("minWithOrThrow-
U")
@ExperimentalUnsignedTypes
@Suppress("CONFLICTING_OVERLOADS")
public fun
ULongArray.minWith(comparator: Comparator<in ULong>): ULong {
 if (isEmpty()) throw NoSuchElementException()
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according
 * to the provided [comparator].
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("minWithOrThrow-
U")
@ExperimentalUnsignedTypes
@Suppress("CONFLICTING_OVERLOADS")
public fun
UByteArray.minWith(comparator: Comparator<in UByte>): UByte {
 if (isEmpty()) throw NoSuchElementException()
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator].
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.7")
@kotlin.jvm.JvmName("minWithOrThrow-
U")
@ExperimentalUnsignedTypes
@Suppress("CONFLICTING_OVERLOADS")
public fun
UShortArray.minWith(comparator: Comparator<in UShort>): UShort {
 if (isEmpty()) throw NoSuchElementException()
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.minWithOrNull(comparator:
Comparator<in UInt>): UInt? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.minWithOrNull(comparator:
Comparator<in ULong>): ULong? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.minWithOrNull(comparator:
Comparator<in UByte>): UByte? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.minWithOrNull(comparator:
Comparator<in UShort>): UShort? {
 if (isEmpty()) return null
 var min = this[0]
 for (i in 1..lastIndex) {
 val e = this[i]
 if (comparator.compare(min, e) > 0) min = e
 }
 return min
}

/** Returns `true` if the array has no elements.
 * @sample samples.collections.Collections.Aggregates.none
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.none(): Boolean {
 return isEmpty()
}

/** Returns `true` if the array has no elements.
 * @sample samples.collections.Collections.Aggregates.none
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.none(): Boolean {
 return isEmpty()
}

/** Returns `true` if the array has no elements.
 * @sample samples.collections.Collections.Aggregates.none
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

```



```

UByteArray.none(): Boolean {
 return isEmpty()
}

Returns `true` if the array has no elements.

@sample samples.collections.Collections.Aggregates.none

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(): Boolean {
 return isEmpty()
}

Returns `true` if no elements match the given [predicate].

@sample samples.collections.Collections.Aggregates.noneWithPredicate

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(predicate: (UInt) -> Boolean): Boolean {
 for (element in this) if (predicate(element)) return false
 return true
}

Returns `true` if no elements match the given [predicate].

@sample samples.collections.Collections.Aggregates.noneWithPredicate

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {
 for (element in this) if (predicate(element)) return false
 return true
}

Returns `true` if no elements match the given [predicate].

@sample samples.collections.Collections.Aggregates.noneWithPredicate

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {
 for (element in this) if (predicate(element)) return false
 return true
}

Returns `true` if no elements match the given [predicate].

@sample samples.collections.Collections.Aggregates.noneWithPredicate

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {
 for (element in this) if (predicate(element)) return false
 return true
}

Returns `true` if no elements match the given [predicate].

@sample samples.collections.Collections.Aggregates.noneWithPredicate

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.onEach(action: (UInt) -> Unit): UIntArray {
 return apply { for (element in this) action(element) }
}

Returns the array itself after performing the given [action] on each element.

@sample samples.collections.Collections.Aggregates.onEach

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.onEach(action: (ULong) -> Unit): ULongArray {
 return apply { for (element in this) action(element) }
}

Returns the array itself after performing the given [action] on each element.

@sample samples.collections.Collections.Aggregates.onEach

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray {
 return apply { for (element in this) action(element) }
}

Returns the array itself after performing the given [action] on each element.

@sample samples.collections.Collections.Aggregates.onEach

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.onEach(action: (UShort) -> Unit): UShortArray {
 return apply { for (element in this) action(element) }
}

Returns the array itself after performing the given [action] on each element, providing sequential index with the element, and returns the array itself afterwards.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

@sample samples.collections.Collections.Aggregates.onEachIndexed

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {
 return apply { forEachIndexed(action) }
}

Returns the array itself after performing the given [action] on each element, providing sequential index with the element, and returns the array itself afterwards.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

@sample samples.collections.Collections.Aggregates.onEachIndexed

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {
 return apply { forEachIndexed(action) }
}

Returns the array itself after performing the given [action] on each element, providing sequential index with the element, and returns the array itself afterwards.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

@sample samples.collections.Collections.Aggregates.onEachIndexed

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {
 return apply { forEachIndexed(action) }
}

Returns the array itself after performing the given [action] on each element, providing sequential index

```

with the element, \n \* and returns the array itself afterwards. \n \* @param [action] function that takes the index of an element and the element itself \n \* and performs the action on the element. \n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {\n return apply {
 forEachIndexed(action) }\n }\n}\n\n/**\n * Accumulates value starting with the
 first element and applying [operation] from left to right\n * to current accumulator value and each element. \n * \n *
 Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use
 [reduceOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes
 current accumulator value and an element, \n * and calculates the next accumulator value. \n * \n * @sample
 samples.collections.Collections.Aggregates.reduce\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {\n if (isEmpty())\n throw
 UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in
 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying
 [operation] from left to right\n * to current accumulator value and each element. \n * \n * Throws an exception if
 this array is empty. If the array can be empty in an expected way, \n * please use [reduceOrNull] instead. It returns
 `null` when its receiver is empty. \n * \n * @param [operation] function that takes current accumulator value and an
 element, \n * and calculates the next accumulator value. \n * \n * @sample
 samples.collections.Collections.Aggregates.reduce\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {\n if (isEmpty())\n throw
 UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in
 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
 right\n * to current accumulator value and each element. \n * \n * Throws an exception if this array is empty. If the
 array can be empty in an expected way, \n * please use [reduceOrNull] instead. It returns `null` when its receiver is
 empty. \n * \n * @param [operation] function that takes current accumulator value and an element, \n * and calculates
 the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {\n if (isEmpty())\n throw
 UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in
 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
 accumulator
 value and each element. \n * \n * Throws an exception if this array is empty. If the array can be empty in an
 expected way, \n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param
 [operation] function that takes current accumulator value and an element, \n * and calculates the next accumulator
 value. \n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduce(operation: (acc: UShort, UShort) -> UShort): UShort {\n if (isEmpty())\n throw
 UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in
 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
 accumulator value and each
 element with its index in the original array. \n * \n * Throws an exception if this array is empty. If the array can be
 empty in an expected way, \n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is
 empty. \n * \n * @param [operation] function that takes the index of an element, current accumulator value and the
```

```

element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt {\n if (isEmpty())\n throw
UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in
1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying
[operation] from left to right\n * to current accumulator value and each element with its index in the original
array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong {\n if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte {\n if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator
= operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element
with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in
an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value and the element itself,\n
* and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort {\n if (isEmpty())\n
throw UnsupportedOperationException("Empty array can't be reduced.")\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator,
this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n
* @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index,
accumulator,
this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the

```

original array.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n \* and calculates the next accumulator value.\n \* \n \* @sample samples.collections.Collections.Aggregates.reduceOrNull\n

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {\n if
(isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element
and applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index
of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte? {\n if
(isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element
with its index in
the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceIndexedOrNull(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort? {\n if
(isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each
element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt?
{\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n
accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with the first element and applying [operation] from left to right\n * to current accumulator value and each
element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the
first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n

```

```

@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceOrNull(operation: (acc:
UByte, UByte) -> UByte): UByte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to
right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceOrNull(operation: (acc: UShort, UShort) ->
UShort): UShort? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex)
{\n accumulator
= operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n
* Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {\n var index = lastIndex\n if (index < 0)
throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator
= operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n
* Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {\n var index = lastIndex\n if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator =
get(index--)\n while (index >= 0) {\n accumulator
= operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n
* Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {\n var index = lastIndex\n if (index
< 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = get(index--)\n
while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please
use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0)
 accumulator = operation(get(index--), accumulator)
 return accumulator
}

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ubyteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {
 var index = lastIndex
 if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
 var accumulator = get(index--)
 while (index >= 0) {
 accumulator = operation(index, get(index), accumulator)
 --index
 }
 return accumulator
}

Accumulates value starting with the last

```

element and applying [operation] from right to left\n \* to each element with its index in the original array and current accumulator value.\n \* \n \* Returns `null` if the array is empty.\n \* \n \* @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n \* and calculates the next accumulator value.\n \* \n

\* @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n

```
\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {\n var index =\n lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull\n

```
\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun\nULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong?\n{\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return\n accumulator\n}\n\n/\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if\n * the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and\n * current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull\n

```
\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\ninline fun\nUByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte?\n{\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return\n accumulator\n}\n\n/\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if\n * the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and\n * current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull\n

```
\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceRightIndexedOrNull(operation: (index:\n Int, UShort, acc: UShort) -> UShort): UShort? {\n var index = lastIndex\n if (index < 0) return null\n var\n accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index),\n accumulator)\n --index\n }\n return accumulator\n}\n\n/\n * Accumulates value starting with the last\n * element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current\n * accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRightOrNull\n

```
\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt):\n UInt? {\n var index = lastIndex\n if (index < 0) return null\n var\n accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/\n * Accumulates value starting with the last element and applying [operation]\n * from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next
```

```

accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) ->
ULong): ULong? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator}\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceRightOrNull(operation: (UByte, acc: UByte) ->
UByte): UByte? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator}\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element
and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightOrNull(operation: (UShort, acc: UShort) ->
UShort): UShort? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator}\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n
* Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.runningFold(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result}\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n
* Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.runningFold(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result}\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with
[initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it
would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator
value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n

```



```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.runningFold(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n *
to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n
* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for
(index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n
}\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n if
(isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
}

```

```

result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R> UShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R>
{\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var
accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
 result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {\n if (isEmpty()) return
emptyList()\n var accumulator = this[0]\n val result = ArrayList<UInt>(size).apply { add(accumulator) }\n for
(index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {\n if (isEmpty()) return
emptyList()\n var accumulator = this[0]\n val result = ArrayList<ULong>(size).apply { add(accumulator) }\n for
(index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element,
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {\n if (isEmpty()) return
emptyList()\n var accumulator = this[0]\n val result = ArrayList<UByte>(size).apply { add(accumulator) }\n for
(index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n
 }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UShortArray.runningReduce(operation: (acc: UShort, UShort) -> UShort): List<UShort> {\n if (isEmpty()) return
emptyList()\n var accumulator = this[0]\n val result = ArrayList<UShort>(size).apply { add(accumulator) }\n
for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n
}\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n
* otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {\n if (isEmpty())
return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UInt>(size).apply { add(accumulator)
}\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current
accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes the index of an element, current accumulator value\n * and the element itself, and
calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {\n if
(isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<ULong>(size).apply {
add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in
the original array and current accumulator value that starts with the first element of this array.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n *
and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {\n if
(isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UByte>(size).apply {
add(accumulator) }\n for (index in 1 until size) {\n
accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return
result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n * to each element, its index in the original array and current accumulator value that starts with the
first
element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.runningReduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): List<UShort> {\n if
(isEmpty())

```

```

return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UShort>(size).apply { add(accumulator)
}\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous
value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element,
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R):
List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R):
List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value
in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scan(initial: R, operation: (acc: R, UShort) -> R):
List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n
*\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R,
UInt) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in
the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value
passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting
list.\n * \n * @param [operation] function that takes the index of an element,

```

```

current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, ULong) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element,
its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc`
value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n *
and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, UByte) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in
the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed
to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n *
\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, UShort) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all
values produced by [selector] function applied to each element in the array.\n * \n * @Deprecated("Use sumOf
instead.", ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince =
"1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumBy(selector: (UInt) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumBy(selector: (ULong) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.sumBy(selector: (UByte) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sumBy(selector: (UShort) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince
= "1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumByDouble(selector: (UInt) -> Double): Double {\n var sum: Double = 0.0\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by

```

```

[selector] function applied to each element in the array.\n *^@Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumByDouble(selector: (ULong) -> Double): Double {\n var sum: Double = 0.0\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n *^@Deprecated("Use sumOf instead.",
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.sumByDouble(selector: (UByte) -> Double): Double {\n var sum: Double = 0.0\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n *^@Deprecated("Use
sumOf instead.", ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sumByDouble(selector: (UShort) -> Double): Double {\n var sum: Double = 0.0\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector:
(UInt) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum
of all values produced by [selector] function applied to each element in the array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector:
(UByte) -> Double): Double {\n
 var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n
 sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*^@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt)
-> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector:
(UByte) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UShortArray.sumOf(selector: (UShort) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n
\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic\n\n inline fun ULongArray.sumOf(selector: (ULong) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n /**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUShort\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic\n\n inline fun UIntArray.sumOf(selector: (UInt) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n /**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic\n\n inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n }\n\n /**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample\n * samples.collections.Iterables.Operations.zipIterable\n */\n\n*\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other: Array<out

```



```

R>): List<Pair<UInt, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Array<out R>): List<Pair<ULong, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other: Array<out
R>): List<Pair<UByte, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other:
Array<out R>): List<Pair<UShort, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
values built from the elements of `this` array and the [other] array with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n
}\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R, V> ULongArray.zip(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n val size
= minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n
}\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n val size = minOf(size,
other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n
}\n return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
infix fun <R> UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> {\n return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample

```

samples.collections.Iterables.Operations.zipIterable

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Iterable<R>): List<Pair<ULong, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other:
Iterable<R>): List<Pair<UByte, R>> {\n return zip(other) { t1,
 t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other:
Iterable<R>): List<Pair<UShort, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
values built from the elements of `this` array and the [other] collection with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Iterable<R>, transform: (a: UInt, b: R) -> V): List<V> {\n val
 arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the
same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has
length of the shortest collection.\n * \n * @sample
```

samples.collections.Iterables.Operations.zipIterableWithTransform

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
ULongArray.zip(other: Iterable<R>, transform: (a: ULong, b: R) -> V): List<V> {\n val arraySize = size\n val
 list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other)
{\n if (i >= arraySize) break\n list.add(transform(this[i++],
 element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the
[other] collection with the same index\n * using the provided [transform] function applied to each pair of
elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
```

samples.collections.Iterables.Operations.zipIterableWithTransform

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> {\n val arraySize = size\n val list
= ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided
```

```
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Iterable<R>, transform: (a: UShort, b: R) -> V): List<V> {\n val arraySize = size\n val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other)
{\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
```

samples.collections.Iterables.Operations.zipIterable

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray.zip(other:
```



```

{\n sum += element\n } \n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic
fun Array<out UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n
}\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sum(): UInt {\n return storage.sum().toUInt()\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sum(): ULong {\n return storage.sum().toULong()\n}\n\n/**\n * Returns the sum of all elements in
the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UByteArray.sum(): UInt {\n return sumOf { it.toUInt() }\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sum(): UInt {\n return sumOf { it.toUInt() }\n}\n\n", "*/\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UCollectionsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n\n/**\n * Returns an array
of UByte containing all of the elements of this collection.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UByte>.toUByteArray():
UByteArray {\n val result = UByteArray(size)\n var index = 0\n for (element in this)\n result[index++] =
element\n return result\n}\n\n/**\n * Returns an array of UInt containing all of the elements of this collection.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UInt>.toUIntArray(): UIntArray
{\n val result = UIntArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n
return result\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this collection.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<ULong>.toULongArray():
ULongArray {\n val result = ULongArray(size)\n var index = 0\n for (element in this)\n result[index++] =
element\n return result\n}\n\n\n/**\n * Returns an array of UShort containing all of the elements of this collection.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UShort>.toUShortArray():
UShortArray {\n val result = UShortArray(size)\n var index = 0\n for (element in this)\n result[index++] =
element\n return result\n}\n\n\n/**\n * Returns the sum of all elements in the collection.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Iterable<UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n\n/**\n * Returns
the sum of all elements in the collection.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n\n/**\n * Returns the sum of all elements in the collection.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Iterable<UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n\n

```

```

sum += element\n }\n return sum\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("\uComparisonsKt")\n\npackage
kotlin.comparisons\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.random.*\n\n/**\n * Returns the greater of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UInt, b:
UInt): UInt {\n return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: ULong,
b: ULong): ULong {\n return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UByte,
b: UByte): UByte {\n return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UShort,
b: UShort): UShort {\n return if (a >= b) a else b\n}\n\n/**\n * Returns
the greater of three values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {\n return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns
the greater of three values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {\n return maxOf(a, maxOf(b, c))\n}\n\n/**\n *
Returns the greater of three values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UByte, b: UByte, c: UByte): UByte {\n return maxOf(a, maxOf(b, c))\n}\n\n/**\n *
Returns the greater of three values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun maxOf(a: UShort, b: UShort, c: UShort): UShort {\n return maxOf(a,
maxOf(b, c))\n}\n\n/**\n * Returns the greater of the given values.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UInt, vararg other: UInt): UInt
{\n var max = a\n for (e in other) max = maxOf(max, e)\n return max\n}\n\n/**\n * Returns the greater of the
given values.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: ULong, vararg
other: ULong): ULong {\n var max = a\n for (e in other) max = maxOf(max, e)\n return max\n}\n\n/**\n *
Returns the greater of the given values.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
maxOf(a: UByte, vararg other: UByte): UByte {\n var max = a\n for (e in other) max = maxOf(max, e)\n
return max\n}\n\n/**\n * Returns the greater of the given values.\n
*\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UShort, vararg other: UShort):
UShort {\n var max = a\n for (e in other) max = maxOf(max, e)\n return max\n}\n\n/**\n * Returns the
smaller of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UInt, b:
UInt): UInt {\n return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: ULong,
b: ULong): ULong {\n return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UByte,
b: UByte): UByte {\n return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UShort,
b: UShort): UShort {\n return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of three values.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline

```

```

fun minOf(a: UInt, b: UInt, c: UInt): UInt {
 return minOf(a, minOf(b, c))
}

Returns the smaller of three values.

@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {
 return minOf(a, minOf(b, c))
}

Returns the smaller of three values.

@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun minOf(a: UByte, b: UByte, c: UByte): UByte {
 return minOf(a, minOf(b, c))
}

Returns the smaller of three values.

@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun minOf(a: UShort, b: UShort, c: UShort): UShort {
 return minOf(a, minOf(b, c))
}

Returns the smaller of the given values.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UInt, vararg
 other: UInt): UInt {
 var min = a
 for (e in other) min = minOf(min, e)
 return min
}

Returns the smaller of the given values.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: ULong, vararg other: ULong): ULong {
 var min = a
 for (e in other) min = minOf(min, e)
 return min
}

Returns the smaller of the given values.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UByte, vararg other: UByte):
 UByte {
 var min = a
 for (e in other) min = minOf(min, e)
 return min
}

Returns the smaller of the given values.

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UShort, vararg
 other: UShort): UShort {
 var min = a
 for (e in other) min = minOf(min, e)
 return min
}

Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("URangesKt")
package
kotlin.ranges
NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
nimport kotlin.random.*
Returns the
first element.
* @throws NoSuchElementException if the progression is empty.

@SinceKotlin("1.7")
public fun UIntProgression.first(): UInt {
 if (isEmpty())
 throw
 NoSuchElementException("Progression $this is empty.")
 return this.first()
}

Returns the first
element.
* @throws NoSuchElementException if the progression is empty.

@SinceKotlin("1.7")
public fun ULongProgression.first(): ULong {
 if (isEmpty())
 throw
 NoSuchElementException("Progression $this is empty.")
 return this.first()
}

Returns the first
element, or `null` if the progression is
empty.

@SinceKotlin("1.7")
public fun UIntProgression.firstOrNull(): UInt? {
 return if (isEmpty())
 null else this.first()
}

Returns the first element, or `null` if the progression is empty.

@SinceKotlin("1.7")
public fun ULongProgression.firstOrNull(): ULong? {
 return if (isEmpty()) null else
 this.first()
}

Returns the last element.
* @throws NoSuchElementException if the progression is
empty.
* @sample samples.collections.Collections.Elements.last

@SinceKotlin("1.7")
public fun
 UIntProgression.last(): UInt {
 if (isEmpty())
 throw NoSuchElementException("Progression $this
 is empty.")
 return this.last()
}

Returns the last element.
* @throws NoSuchElementException if
the progression is empty.
* @sample samples.collections.Collections.Elements.last

@SinceKotlin("1.7")
public fun ULongProgression.last(): ULong {
 if (isEmpty())
 throw
 NoSuchElementException("Progression
 $this is empty.")
 return this.last()
}

Returns the last element, or `null` if the progression is empty.

* @sample samples.collections.Collections.Elements.last

@SinceKotlin("1.7")
public fun
 UIntProgression.lastOrNull(): UInt? {
 return if (isEmpty()) null else this.last()
}

Returns the last
element, or `null` if the progression is empty.
* @sample samples.collections.Collections.Elements.last

@SinceKotlin("1.7")
public fun ULongProgression.lastOrNull(): ULong? {
 return if (isEmpty()) null else

```

```

this.last()\n\n/**\n * Returns a random element from this range.\n * \n * @throws IllegalArgumentException if this
range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun UIntRange.random(): UInt {\n return random(Random)\n}\n\n/**\n * Returns a random element
from this range.\n * \n * @throws IllegalArgumentException if this
range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun ULongRange.random(): ULong {\n return random(Random)\n}\n\n/**\n * Returns a random
element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if
this range is empty.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n
npublic fun UIntRange.random(random: Random): UInt {\n try {\n return random.nextUInt(this)\n }
catch(e:
IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness.\n * \n * @throws
IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n
npublic fun ULongRange.random(random: Random): ULong {\n try {\n return random.nextULong(this)\n
 }
catch(e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n *
Returns a random element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun UIntRange.randomOrNull():
UInt? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if
this
range is empty.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline fun ULongRange.randomOrNull():
ULong? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n
npublic fun UIntRange.randomOrNull(random:
Random): UInt? {\n if (isEmpty())\n return null\n return random.nextUInt(this)\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n
npublic fun ULongRange.randomOrNull(random: Random): ULong? {\n if
(isEmpty())\n return null\n return random.nextULong(this)\n}\n\n/**\n * Returns `true` if this range contains
the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic inline operator fun UIntRange.contains(element: UInt?): Boolean {\n return element != null &&
contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always
returns
`false` if the [element] is `null`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
npublic
inline operator fun ULongRange.contains(element: ULong?): Boolean {\n return element != null &&
contains(element)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n
npublic operator fun
UIntRange.contains(value: UByte): Boolean {\n return contains(value.toUInt())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n
npublic operator fun
ULongRange.contains(value: UByte): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n
npublic operator fun

```

```

ULongRange.contains(value: UInt): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nUIntRange.contains(value: ULong): Boolean {\n return (value shr UInt.SIZE_BITS) == 0uL &&\n contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nUIntRange.contains(value: UShort): Boolean {\n return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nULongRange.contains(value: UShort): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUByte.downTo(to: UByte): UIntProgression {\n return UIntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUInt.downTo(to: UInt): UIntProgression {\n return UIntProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nULong.downTo(to: ULong): ULongProgression {\n return ULongProgression.fromClosedRange(this, to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUShort.downTo(to: UShort): UIntProgression {\n return UIntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun\nUIntProgression.reversed(): UIntProgression {\n return UIntProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun\nULongProgression.reversed(): ULongProgression {\n return ULongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUIntProgression.step(step: Int): UIntProgression {\n checkStepIsPositive(step > 0, step)\n return UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nULongProgression.step(step: Long): ULongProgression {\n checkStepIsPositive(step > 0, step)\n return ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUByte.until(to: UByte): UIntRange {\n if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY\n return

```



```

this.toUInt() .. (to - 1u).toUInt()\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UInt.until(to:
UInt): UIntRange {\n if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n return this .. (to -
1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or
equal to `this` value, then the returned range is empty.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULong.until(to: ULong): ULongRange {\n if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY\n
return this .. (to - 1u).toULong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UShort.until(to: UShort): UIntRange {\n if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY\n return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Ensures that this value is not less than the specified
[minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the
[minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun UInt.coerceAtLeast(minimumValue: UInt): UInt {\n return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtLeast(minimumValue: ULong): ULong {\n return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun UByte.coerceAtLeast(minimumValue: UByte): UByte {\n return if (this < minimumValue) minimumValue
else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this
value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtLeast(minimumValue: UShort): UShort {\n return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtMost(maximumValue:
UInt): UInt {\n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is
not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the
[maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtMost(maximumValue: ULong): ULong {\n return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun

```

```

UByte.coerceAtMost(maximumValue: UByte): UByte {
 return if (this > maximumValue) maximumValue
 else this
}
/**
 * Ensures that this value is not greater than the specified [maximumValue].
 * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.
 * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public fun
UShort.coerceAtMost(maximumValue: UShort): UShort {
 return if (this > maximumValue) maximumValue
 else this
}
/**
 * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or
 * [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public fun
UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {
 if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
 $maximumValue is less than minimum $minimumValue.")
 if (this < minimumValue) return minimumValue
 if (this > maximumValue) return maximumValue
 return this
}
/**
 * Ensures that this value lies in the
 * specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or
 * [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
 * [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public fun
ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {
 if (minimumValue >
 maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
 $maximumValue is less than minimum $minimumValue.")
 if (this < minimumValue) return minimumValue
 if (this > maximumValue) return maximumValue
 return this
}
/**
 * Ensures that this value lies in the
 * specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or
 * [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
 * [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public fun
UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
 if (minimumValue >
 maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
 $maximumValue is less than minimum $minimumValue.")
 if (this < minimumValue) return minimumValue
 if (this > maximumValue) return maximumValue
 return this
}
/**
 * Ensures that this value lies in the
 * specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range,
 * or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
 * [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public fun
UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {
 if (minimumValue >
 maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
 $maximumValue is less than minimum $minimumValue.")
 if (this < minimumValue) return minimumValue
 if (this > maximumValue) return maximumValue
 return this
}
/**
 * Ensures that this value lies in the
 * specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than
 * `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
public
fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {
 if (range is ClosedFloatingPointRange) {
 return
 this.coerceIn<UInt>(range)
 }
 if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to
 an empty range: $range.")
 return when {
 this < range.start -> range.start
 this > range.endInclusive -
 > range.endInclusive
 else -> this
 }
}
/**
 * Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or
 */

```

```

`range.endInclusive` if this value is greater than `range.endInclusive`. \n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(range: ClosedRange<ULong>): ULong {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<ULong>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n@\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("USequencesKt")\n\npackage
kotlin.sequences\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the
sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic
fun Sequence<UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this)
{\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n *
The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic
fun Sequence<UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n
 }\n return sum\n}\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin\n\npublic expect open class Error : Throwable {\n
 constructor()\n constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n
 constructor(cause: Throwable?)\n}\n\npublic expect open class Exception : Throwable {\n constructor()\n
 constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class RuntimeException : Exception {\n constructor()\n constructor(message:
String?)\n constructor(message: String?, cause: Throwable?)\n}\n\npublic expect open class
IllegalStateException : RuntimeException {\n constructor()\n constructor(message: String?)\n
 constructor(message: String?, cause: Throwable?)\n}\n\npublic expect open class
IllegalStateException : RuntimeException {\n constructor()\n constructor(message: String?)\n
 constructor(message: String?, cause: Throwable?)\n}\n\npublic expect open class
IndexOutOfBoundsException : RuntimeException {\n constructor()\n constructor(message:
String?)\n}\n\npublic expect open class ConcurrentModificationException : RuntimeException {\n constructor()\n
 constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class UnsupportedOperationException : RuntimeException {\n
 constructor()\n constructor(message: String?)\n}

```



mangling.

- \* This annotation can be applied to either files or top-level declarations.
- \* It is currently prohibited to export the following kinds of declarations:
  - \* `expect` declarations
  - \* inline functions with reified type parameters
  - \* suspend functions
  - \* secondary constructors without `@JsName`
  - \* extension properties
  - \* enum classes
  - \* annotation classes
- \* Signatures of exported declarations must only contain "exportable" types:
  - \* `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`
  - \* `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`
  - \* `Array<exportable-type>`
  - \* Function types with exportable parameters and return types
  - \* `external` or `@JsExport` classes and interfaces
  - \* Nullable counterparts of types above
  - \* Unit return type. Must not be nullable
- \* This annotation is experimental, meaning that restrictions mentioned above are subject to change.

```

*\/@ExperimentalJsExport@Retention(AnnotationRetention.BINARY)@Target(CLASS,
PROPERTY, FUNCTION, FILE)@SinceKotlin("1.4")@OptionalExpectation
public expect annotation class JsExport() {
 /* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
 package kotlin.io
 /** Prints the line separator to the standard output stream.
 */
 public expect fun println()
 /** Prints the given [message] and the line separator to the standard output stream.
 */
 public expect fun println(message: Any?)
 /** Prints the given [message] to the standard output stream.
 */
 public expect fun print(message: Any?)
 /** Reads a line of input from the standard input stream and returns it,
 or throws a [RuntimeException] if EOF has already been reached when [readln] is called.
 LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.
 Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].
 */
 @SinceKotlin("1.6")
 public expect fun readln(): String
 /** Reads a line of input from the standard input stream and returns it,
 or return `null` if EOF has already been reached when [readlnOrNull] is called.
 LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.
 Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].
 */
 @SinceKotlin("1.6")
 public expect fun readlnOrNull(): String?
}
internal class ReadAfterEOFException(message: String?) :
 RuntimeException(message)
internal expect interface Serializable {
 /* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
 package kotlin.collections
 import kotlin.internal.PlatformDependent
 /**
 * Classes that inherit from this interface can be represented as a sequence of elements that can
 * be iterated over.
 * @param T the type of element being iterated over. The iterator is covariant in its element type.
 */
 public interface Iterable<out T> {
 /**
 * Returns an iterator over the elements of this object.
 */
 public operator fun iterator(): Iterator<T>
 }
 /**
 * Classes that inherit from this interface can be represented as a sequence of elements that can
 * be iterated over and that supports removing elements during iteration.
 * @param T the type of element being iterated over. The mutable iterator is invariant in its element type.
 */
 public interface MutableIterable<out T> : Iterable<T> {
 /**
 * Returns an iterator over the elements of this sequence that supports removing elements during iteration.
 */
 override fun iterator(): MutableIterator<T>
 }
 /**
 * A generic collection of elements. Methods in this interface support only read-only access to the collection;
 * read/write access is supported through the [MutableCollection] interface.
 * @param E the type of elements contained in the collection. The collection is covariant in its element type.
 */
 public interface Collection<out E> : Iterable<E> {
 // Query Operations
 /**
 * Returns the size of the collection.
 */
 public val size: Int
 /**
 * Returns `true` if the collection is empty (contains no elements), `false` otherwise.
 */
 public fun isEmpty(): Boolean
 /**
 * Checks if the specified element is contained in this collection.
 */
 public operator fun contains(element: @UnsafeVariance E): Boolean
 /**
 * override fun iterator(): Iterator<E>
 // Bulk Operations
 /**
 * Checks if all elements in the specified collection are contained in this collection.
 */
 public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
 }
}

```

```

* A generic collection of elements that supports adding and removing elements.\n *\n * @param E the type of
elements contained in the collection. The mutable collection is invariant in its element type.\n */\npublic interface
MutableCollection<E> : Collection<E>, MutableIterable<E> {\n // Query Operations\n override fun iterator():
MutableIterator<E>\n\n // Modification Operations\n /**\n * Adds the specified element to the collection.\n
*\n * @return `true` if the element has been added, `false` if the collection does not support duplicates\n * and
the element is already contained in the collection.\n */\n public fun add(element: E): Boolean\n\n /**\n *
Removes a single instance of the specified element from this\n * collection, if it is present.\n *\n * @return
`true` if the element has been successfully removed; `false` if it was not present in the collection.\n */\n public
fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n /**\n * Adds all of the elements of the specified collection to this
collection.\n *\n * @return `true` if any of the specified elements was added to the collection, `false` if the
collection was not modified.\n */\n public fun addAll(elements: Collection<E>): Boolean\n\n /**\n *
Removes all of this collection's elements that are also contained in the specified collection.\n *\n * @return
`true` if any of the specified elements was removed from the collection, `false` if the collection was not modified.\n
*/\n public fun removeAll(elements: Collection<E>): Boolean\n\n /**\n * Retains only the elements in this
collection that are contained in the specified collection.\n *\n * @return `true` if any element was removed
from the collection, `false` if the collection was not modified.\n */\n public fun retainAll(elements:
Collection<E>): Boolean\n\n /**\n * Removes all elements
from this collection.\n */\n public fun clear(): Unit\n}\n\n/**\n * A generic ordered collection of elements.
Methods in this interface support only read-only access to the list;\n * read/write access is supported through the
[MutableList] interface.\n * @param E the type of elements contained in the list. The list is covariant in its element
type.\n */\npublic interface List<out E> : Collection<E> {\n // Query Operations\n\n override val size: Int\n
override fun isEmpty(): Boolean\n override fun contains(element: @UnsafeVariance E): Boolean\n override fun
iterator(): Iterator<E>\n\n // Bulk Operations\n override fun containsAll(elements: Collection<@UnsafeVariance
E>): Boolean\n\n // Positional Access Operations\n /**\n * Returns the element at the specified index in the
list.\n */\n public operator fun get(index: Int): E\n\n // Search Operations\n /**\n * Returns the index of
the first occurrence of the specified element in the list,
or -1 if the specified\n * element is not contained in the list.\n */\n public fun indexOf(element:
@UnsafeVariance E): Int\n\n /**\n * Returns the index of the last occurrence of the specified element in the list,
or -1 if the specified\n * element is not contained in the list.\n */\n public fun lastIndexOf(element:
@UnsafeVariance E): Int\n\n // List Iterators\n /**\n * Returns a list iterator over the elements in this list (in
proper sequence).\n */\n public fun listIterator(): ListIterator<E>\n\n /**\n * Returns a list iterator over the
elements in this list (in proper sequence), starting at the specified [index].\n */\n public fun listIterator(index:
Int): ListIterator<E>\n\n // View\n /**\n * Returns a view of the portion of this list between the specified
[fromIndex] (inclusive) and [toIndex] (exclusive).\n * The returned list is backed by this list, so non-structural
changes in the returned list are
reflected in this list, and vice-versa.\n *\n * Structural changes in the base list make the behavior of the view
undefined.\n */\n public fun subList(fromIndex: Int, toIndex: Int): List<E>\n}\n\n/**\n * A generic ordered
collection of elements that supports adding and removing elements.\n * @param E the type of elements contained in
the list. The mutable list is invariant in its element type.\n */\npublic interface MutableList<E> : List<E>,
MutableCollection<E> {\n // Modification Operations\n /**\n * Adds the specified element to the end of this
list.\n *\n * @return `true` because the list is always modified as the result of this operation.\n */\n override
fun add(element: E): Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification
Operations\n /**\n * Adds all of the elements of the specified collection to the end of this list.\n *\n * The
elements are appended in the order they appear in the [elements]
collection.\n *\n * @return `true` if the list was changed as the result of the operation.\n */\n override fun
addAll(elements: Collection<E>): Boolean\n\n /**\n * Inserts all of the elements of the specified collection
[elements] into this list at the specified [index].\n *\n * @return `true` if the list was changed as the result of the

```

```

operation.\n */\n public fun addAll(index: Int, elements: Collection<E>): Boolean\n\n override fun\n removeAll(elements: Collection<E>): Boolean\n\n override fun retainAll(elements: Collection<E>): Boolean\n\n override fun clear(): Unit\n\n // Positional Access Operations\n\n /**\n * Replaces the element at the specified\n * position in this list with the specified element.\n *\n * @return the element previously at the specified\n * position.\n */\n public operator fun set(index: Int, element: E): E\n\n /**\n * Inserts an element into the list\n * at the specified [index].\n */\n public\n fun add(index: Int, element: E): Unit\n\n /**\n * Removes an element at the specified [index] from the list.\n *\n * @return the element that has been removed.\n */\n public fun removeAt(index: Int): E\n\n // List\n Iterators\n\n override fun listIterator(): MutableListIterator<E>\n\n override fun listIterator(index: Int):\n MutableListIterator<E>\n\n // View\n\n override fun subList(fromIndex: Int, toIndex: Int):\n MutableList<E>\n\n /**\n * A generic unordered collection of elements that does not support duplicate\n * elements.\n * Methods in this interface support only read-only access to the set;\n * read/write access is supported\n * through the [MutableSet] interface.\n * @param E the type of elements contained in the set. The set is covariant in\n * its element type.\n */\n public interface Set<out E> : Collection<E> {\n\n // Query Operations\n\n override val size:\n Int\n\n override fun isEmpty(): Boolean\n\n override fun contains(element: @UnsafeVariance\n E): Boolean\n\n override fun iterator(): Iterator<E>\n\n // Bulk Operations\n\n override fun containsAll(elements:\n Collection<@UnsafeVariance E>): Boolean\n\n /**\n * A generic unordered collection of elements that does not\n * support duplicate elements, and supports\n * adding and removing elements.\n * @param E the type of elements\n * contained in the set. The mutable set is invariant in its element type.\n */\n public interface MutableSet<E> : Set<E>,\n MutableCollection<E> {\n\n // Query Operations\n\n override fun iterator(): MutableIterator<E>\n\n //\n Modification Operations\n\n /**\n * Adds the specified element to the set.\n *\n * @return `true` if the\n * element has been added, `false` if the element is already contained in the set.\n */\n override fun add(element:\n E): Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n\n override\n fun addAll(elements: Collection<E>): Boolean\n\n override fun removeAll(elements:\n Collection<E>): Boolean\n\n override fun retainAll(elements: Collection<E>): Boolean\n\n override fun clear():\n Unit\n\n /**\n * A collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n * the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * Methods\n * in this interface support only read-only access to the map; read-write access is supported through\n * the\n * [MutableMap] interface.\n * @param K the type of map keys. The map is invariant in its key type, as it\n * can\n * accept key as a parameter (of [containsKey] for example) and return it in [keys] set.\n * @param V the type of map\n * values. The map is covariant in its value type.\n */\n public interface Map<K, out V> {\n\n // Query Operations\n\n /**\n * Returns the number of key/value pairs in the map.\n */\n public val size: Int\n\n /**\n * Returns\n * `true` if the map is empty (contains no elements), `false` otherwise.\n */\n public fun isEmpty(): Boolean\n\n /**\n * Returns `true` if the map contains the specified [key].\n */\n public fun containsKey(key: K): Boolean\n\n /**\n * Returns `true` if the map maps one or more keys to\n * the specified [value].\n */\n public fun containsValue(value: @UnsafeVariance V): Boolean\n\n /**\n * Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.\n */\n public operator fun get(key: K): V?\n\n /**\n * Returns the value corresponding to the given [key], or\n * [defaultValue] if such a key is not present in the map.\n *\n * @since JDK 1.8\n */\n @SinceKotlin("1.1")\n @PlatformDependent\n public fun getOrDefault(key: K, defaultValue:\n @UnsafeVariance V): V {\n\n // See default implementation in JDK sources\n throw\n NotImplementedError()\n }\n\n // Views\n\n /**\n * Returns a read-only [Set] of all keys in this map.\n */\n public val keys: Set<K>\n\n /**\n * Returns a read-only [Collection] of all values in this map. Note that this\n * collection may contain duplicate values.\n */\n public val values: Collection<V>\n\n /**\n * Returns a read-\n * only [Set] of all key/value pairs in this map.\n */\n public val entries: Set<Map.Entry<K, V>>\n\n /**\n * Represents a key/value pair held by a [Map].\n */\n public interface Entry<out K, out V> {\n\n /**\n * Returns the key of this key/value pair.\n */\n public val key: K\n\n /**\n * Returns the value of\n * this key/value pair.\n */\n public val value: V\n }\n\n /**\n * A modifiable collection that holds pairs

```

of objects (keys and values) and supports efficiently retrieving the value corresponding to each key. Map keys are unique; the map holds only one value for each key. @param K the type of map keys. The map is invariant in its key type. @param V the

type of map values. The mutable map is invariant in its value type. `public interface MutableMap<K, V> : Map<K, V> {`  
// Modification Operations  
/\*\* Associates the specified [value] with the specified [key] in the map.  
\* @return the previous value associated with the key, or `null` if the key was not present in the map.  
\*/  
public fun put(key: K, value: V): V?  
/\*\* Removes the specified key and its corresponding value from this map.  
\* @return the previous value associated with the key, or `null` if the key was not present in the map.  
\*/  
public fun remove(key: K): V?  
/\*\* Removes the entry for the specified key only if it is mapped to the specified value.  
\* @return true if entry was removed  
\*/  
@SinceKotlin("1.1")  
@PlatformDependent  
public fun remove(key: K, value: V): Boolean {  
// See default implementation in JDK sources  
return true  
}

// Bulk Modification Operations  
/\*\* Updates this map with key/value pairs from the specified map [from].  
\*/  
public fun putAll(from: Map<out K, V>): Unit  
/\*\* Removes all elements from this map.  
\*/  
public fun clear(): Unit  
// Views  
/\*\* Returns a [MutableSet] of all keys in this map.  
\*/  
override val keys: MutableSet<K>  
/\*\* Returns a [MutableCollection] of all values in this map. Note that this collection may contain duplicate values.  
\*/  
override val values:

MutableCollection<V>  
/\*\* Returns a [MutableSet] of all key/value pairs in this map.  
\*/  
override val entries: MutableSet<MutableMap.MutableEntry<K, V>>  
/\*\* Represents a key/value pair held by a [MutableMap].  
\*/  
public interface MutableEntry<K, V> : Map.Entry<K, V> {  
/\*\* Changes the value associated with the key of this entry.  
\*/  
\* @return the previous value corresponding to the key.  
\*/  
public fun setValue(newValue: V): V

}  
}"}  
/\* Copyright 2010-2015 JetBrains s.r.o.  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\* http://www.apache.org/licenses/LICENSE-2.0  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/  
package kotlin

public object Unit {  
// The type with only one value: the `Unit` object. This type corresponds to the `void` type in Java.  
// public object  
override fun toString() = "kotlin.Unit"  
}"}  
/\* Copyright 2010-2015 JetBrains s.r.o.  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\* http://www.apache.org/licenses/LICENSE-2.0  
\* Unless required by applicable law or agreed to in writing,  
software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/  
package kotlin.annotation

import kotlin.annotation.AnnotationTarget.  
/\*\* Contains the list of code elements which are the possible annotation targets.  
\*/  
public enum class AnnotationTarget {  
/\*\* Class, interface or object, annotation class is also included  
\*/  
CLASS,  
/\*\* Annotation class only  
\*/  
ANNOTATION\_CLASS,  
/\*\* Generic type parameter  
\*/  
TYPE\_PARAMETER,  
/\*\* Property  
\*/  
PROPERTY,  
/\*\* Field, including property's backing field  
\*/  
FIELD,  
/\*\* Local variable  
\*/  
LOCAL\_VARIABLE,  
/\*\* Value parameter of a function or a constructor  
\*/  
VALUE\_PARAMETER,  
/\*\* Constructor only (primary or secondary)  
\*/  
CONSTRUCTOR,  
/\*\* Function (constructors are not included)  
\*/  
FUNCTION,  
/\*\* Property getter only  
\*/  
PROPERTY\_GETTER,  
/\*\* Property setter only  
\*/  
PROPERTY\_SETTER,  
/\*\* Type usage  
\*/  
TYPE,  
/\*\* Any expression  
\*/  
EXPRESSION,  
/\*\* File  
\*/  
FILE,  
/\*\* Type alias  
\*/  
@SinceKotlin("1.1")  
TYPEALIAS

/\*\* Contains the list of possible annotation's retentions.  
\*/  
\* Determines how an annotation is stored in binary output.  
\*/  
public enum class AnnotationRetention {  
/\*\* Annotation isn't stored



```

in binary output */\n SOURCE,\n /** Annotation is stored in binary output, but invisible for reflection */\n
BINARY,\n /** Annotation is stored
in binary output and visible for reflection (default retention) */\n RUNTIME\n}\n\n/**\n * This meta-annotation
indicates the kinds of code elements which are possible targets of an annotation.\n *\n * If the target meta-annotation
is not present on an annotation declaration, the annotation is applicable to the following elements:\n * [CLASS],
[PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION],
[PROPERTY_GETTER], [PROPERTY_SETTER].\n *\n * @property allowedTargets list of allowed annotation
targets\n */\n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@MustBeDocumented\npublic annotation
class Target(vararg val allowedTargets: AnnotationTarget)\n\n/**\n * This meta-annotation determines whether an
annotation is stored in binary output and visible for reflection. By default, both are true.\n *\n * @property value
necessary annotation retention (RUNTIME, BINARY or SOURCE)\n
*/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Retention(val
value: AnnotationRetention = AnnotationRetention.RUNTIME)\n\n/**\n * This meta-annotation determines that an
annotation is applicable twice or more on a single code element\n
*/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Repeatable\n\n/**\n * This
meta-annotation determines that an annotation is a part of public API and therefore should be included in the
generated\n * documentation for the element to which the annotation is applied.\n
/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class MustBeDocumented\n","/\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@JsName("arrayIterator")\ninternal fun arrayIterator(array: dynamic, type: String?) = when (type) {\n null
-> {\n val arr: Array<dynamic> = array\n object : Iterator<dynamic> {\n var index =
0\n override fun hasNext() = index < arr.size\n override fun next() = if (index < arr.size) arr[index++]
else throw NoSuchElementException("$index")\n }
}\n "BooleanArray" ->
booleanArrayIterator(array)\n "ByteArray" -> byteArrayIterator(array)\n "ShortArray" ->
shortArrayIterator(array)\n "CharArray" -> charArrayIterator(array)\n "IntArray" -> intArrayIterator(array)\n
"LongArray" -> longArrayIterator(array)\n "FloatArray" -> floatArrayIterator(array)\n "DoubleArray" ->
doubleArrayIterator(array)\n else -> throw IllegalStateException("Unsupported type argument for arrayIterator:
$type")\n}\n\n@JsName("booleanArrayIterator")\ninternal fun booleanArrayIterator(array: BooleanArray) =
object : BooleanIterator() {\n var index = 0\n override fun hasNext() = index < array.size\n override fun
nextBoolean() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("byteArrayIterator")\ninternal
fun byteArrayIterator(array: ByteArray) = object : ByteIterator() {\n var index = 0\n override fun hasNext() =
index < array.size\n override fun nextByte() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("shortArrayIterator")\ninternal fun
shortArrayIterator(array: ShortArray) = object : ShortIterator() {\n var index = 0\n override fun hasNext() =
index < array.size\n override fun nextShort() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("charArrayIterator")\ninternal fun charArrayIterator(array:
CharArray) = object : CharIterator() {\n var index = 0\n override fun hasNext() = index < array.size\n override
fun nextChar() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("intArrayIterator")\ninternal fun intArrayIterator(array:
IntArray) = object : IntIterator()
{\n var index = 0\n override fun hasNext() = index < array.size\n override fun nextInt() = if (index <
array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("floatArrayIterator")\ninternal fun
floatArrayIterator(array: FloatArray) = object : FloatIterator() {\n var index = 0\n override fun hasNext() = index
< array.size\n override fun nextFloat() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("doubleArrayIterator")\ninternal fun

```

```

doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {
 var index = 0
 override fun hasNext() = index < array.size
 override fun nextDouble() = if (index < array.size) array[index++] else throw
 NoSuchElementException("$index")
}
@JsName("longArrayIterator")
internal fun longArrayIterator(array: LongArray) = object : LongIterator() {
 var index = 0
 override fun hasNext() = index < array.size
 override fun
 nextLong() = if (index < array.size) array[index++] else throw
 NoSuchElementException("$index")
}
@JsName("PropertyMetadata")
internal class PropertyMetadata(@JsName("callableName") val name: String)
@JsName("noWhenBranchMatched")
internal fun noWhenBranchMatched(): Nothing = throw
 NoWhenBranchMatchedException()
@JsName("subSequence")
internal fun subSequence(c: CharSequence, startIndex: Int, endIndex: Int): CharSequence {
 if (c is String) {
 return c.substring(startIndex, endIndex)
 } else {
 return c.asDynamic().`subSequence_vux9f0$(startIndex, endIndex)`
 }
}
@JsName("captureStack")
internal fun captureStack(@Suppress("UNUSED_PARAMETER") baseClass: JsClass<in Throwable>, instance: Throwable) {
 if (js("Error").captureStackTrace) {
 // Using uncropped stack traces due to KT-37563.
 // Precise stack traces are implemented in JS IR compiler and stdlib
 js("Error").captureStackTrace(instance);
 } else {
 instance.asDynamic().stack = js("new Error()").stack;
 }
}
@JsName("newThrowable")
internal fun newThrowable(message: String?, cause: Throwable?): Throwable {
 val throwable = js("new Error()")
 throwable.message = if (jsTypeOf(message) == "undefined") {
 if (cause != null) cause.toString() else null
 } else {
 message
 }
 throwable.cause = cause
 throwable.name = "Throwable"
 return throwable
}
@JsName("BoxedChar")
internal class BoxedChar(val c: Int) : Comparable<Int> {
 override fun equals(other: Any?): Boolean {
 return other is BoxedChar && c == other.c
 }
 override fun hashCode(): Int {
 return c
 }
 override fun toString(): String {
 return js("this.c").unsafeCast<Char>().toString()
 }
 override fun compareTo(other: Int): Int {
 return js("this.c - other").unsafeCast<Int>()
 }
}
@JsName("valueOf")
public fun valueOf(): Int {
 return c
}
@kotlin.internal.InlineOnly
internal inline fun <T> concat(args: Array<T>): T {
 val typed = js("Array")(args.size)
 for (i in args.indices) {
 val arr = args[i]
 if (arr !is Array<*>) {
 typed[i] = js("[]").slice.call(arr)
 } else {
 typed[i] = arr
 }
 }
 return js("[]").concat.apply(js("[]"), typed);
}
/** Concat regular Array's and TypedArray's into an Array.
 *
 * @PublishedApi
 * @JsName("arrayConcat")
 * @Suppress("UNUSED_PARAMETER")
 * internal fun <T> arrayConcat(a: T, b: T): T {
 return concat(js("arguments"))
}
/** Concat primitive arrays. Main use:
 *
 * prepare vararg arguments.
 * For compatibility with 1.1.0 the arguments may be a mixture of Array's and TypedArray's.
 * If the first argument is TypedArray (Byte-, Short-, Char-, Int-, Float-, and DoubleArray) returns a TypedArray, otherwise an Array.
 * If the first argument has the $type$ property (Boolean-, Char-, and LongArray) copy its value to result.$type$.
 * If the first argument is a regular Array without the $type$ property default to arrayConcat.
 *
 * @PublishedApi
 * @JsName("primitiveArrayConcat")
 * @Suppress("UNUSED_PARAMETER")
 * internal fun <T> primitiveArrayConcat(a: T, b: T): T {
 val args: Array<T> = js("arguments")
 if (a is Array<*> && a.asDynamic().`$type$` === undefined) {
 return concat(args)
 } else {
 var size = 0
 for (i in args.indices) {
 size += args[i].asDynamic().length as Int
 }
 val result = js("new a.constructor(size)")
 kotlin.copyArrayType(a, result)
 size = 0
 for (i in args.indices) {
 val arr = args[i].asDynamic()
 for (j in 0 until arr.length) {
 result[size++] = arr[j]
 }
 }
 return result
 }
}
@JsName("booleanArrayOf")
internal fun booleanArrayOf() = withType("BooleanArray", js("[]").slice.call(arguments))
@JsName("charArrayOf")
internal fun charArrayOf() = withType("CharArray", js("new Uint16Array(arguments)"))
@JsName("longArrayOf")
internal fun longArrayOf() = withType("LongArray",

```

```

js("[].slice.call(arguments)")\n\n@JsName("withType")\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun
withType(type: String, array: dynamic): dynamic {\n array.`$type$` = type\n return array\n}\n\n"/*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n *
Function corresponding to JavaScript's `typeof` operator\n
*/\n\n@kotlin.internal.InlineOnly\n\n@Suppress("UNUSED_PARAMETER")\n\npublic inline fun jsTypeOf(a: Any?):
String = js("typeof a")\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:Suppress("UNUSED_PARAMETER", "NOTHING_TO_INLINE")\n\npackage kotlin\n\n/**\n *
Returns an empty array of the specified type [T].\n */\n\npublic inline fun <T> emptyArray(): Array<T> =
js("[]")\n\n\n@library\n\npublic fun <T> arrayOf(vararg elements: T): Array<T> =
definedExternally\n\n\n@library\n\npublic fun doubleArrayOf(vararg elements: Double): DoubleArray =
definedExternally\n\n\n@library\n\npublic fun floatArrayOf(vararg elements: Float): FloatArray =
definedExternally\n\n\n@library\n\npublic fun longArrayOf(vararg elements: Long): LongArray =
definedExternally\n\n\n@library\n\npublic fun intArrayOf(vararg elements: Int): IntArray =
definedExternally\n\n\n@library\n\npublic fun charArrayOf(vararg elements: Char): CharArray =
definedExternally\n\n\n@library\n\npublic fun shortArrayOf(vararg elements: Short): ShortArray =
definedExternally\n\n\n@library\n\npublic fun byteArrayOf(vararg
elements: Byte): ByteArray = definedExternally\n\n\n@library\n\npublic fun booleanArrayOf(vararg elements:
Boolean): BooleanArray = definedExternally\n\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified
initialization function [initializer].\n */\n\npublic actual fun <T> lazy(initializer: () -> T): Lazy<T> =
UnsafeLazyImpl(initializer)\n\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization
function [initializer].\n * The [mode] parameter is ignored.\n */\n\npublic actual fun <T> lazy(mode:
LazyThreadSafetyMode, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n\n/**\n * Creates a new
instance of the [Lazy] that uses the specified initialization function [initializer].\n * The [lock] parameter is
ignored.\n */\n\npublic actual fun <T> lazy(lock: Any?, initializer: () -> T): Lazy<T> =
UnsafeLazyImpl(initializer)\n\n\ninternal fun fillFrom(src: dynamic, dst: dynamic): dynamic {\n val srcLen: Int =
src.length\n val
dstLen: Int = dst.length\n var index: Int = 0\n while (index < srcLen && index < dstLen) dst[index] =
src[index++]\n return dst\n}\n\n\ninternal fun arrayCopyResize(source: dynamic, newSize: Int, defaultValue:
Any?): dynamic {\n val result = source.slice(0, newSize)\n copyArrayType(source, result)\n var index: Int =
source.length\n if (newSize > index) {\n result.length = newSize\n while (index < newSize)
result[index++] = defaultValue\n }\n return result\n}\n\n\ninternal fun <T> arrayPlusCollection(array: dynamic,
collection: Collection<T>): dynamic {\n val result = array.slice()\n result.length += collection.size\n copyArrayType(array, result)\n var index: Int = array.length\n for (element in collection) result[index++] =
element\n return result\n}\n\n\ninternal fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection:
Collection<T>): dynamic {\n var index = startIndex\n for (element in collection) dst[index++]
= element\n return dst\n}\n\n\ninternal inline fun copyArrayType(from: dynamic, to: dynamic) {\n if
(from.`$type$` !== undefined) {\n to.`$type$` = from.`$type$`\n }\n}\n\n\ninternal inline fun jsIsType(obj:
dynamic, jsClass: dynamic) = js("Kotlin").isType(obj, jsClass)","/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Creates a Char with the specified
[code].\n */\n\n * @sample samples.text.Chars.charFromCode\n
*/\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\n@kotlin.internal.InlineOnly\n\npublic
actual inline fun Char(code: UShort): Char {\n return code.toInt().toChar()\n}\n\n"/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt

```

```

file.\n *\n\npackage kotlin.coroutines\n\nimport
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@SinceKotlin("1.3")\n\n@JsName("CoroutineImpl")\n
ninternal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?>
{\n protected var state = 0\n protected var exceptionState = 0\n protected var result: Any? = null\n protected
var exception: Throwable? = null\n protected var finallyPath: Array<Int>? = null\n\n public override val context:
CoroutineContext = resultContinuation.context\n\n private var intercepted_: Continuation<Any?>? = null\n\n
public fun intercepted(): Continuation<Any?> =\n intercepted_\n ?:\n
(context[ContinuationInterceptor]?.interceptContinuation(this) ?: this)\n .also { intercepted_ = it }\n\n
override fun resumeWith(result: Result<Any?>) {\n var current = this\n var currentResult: Any? =
result.getOrNull()\n var currentException:
Throwable? = result.exceptionOrNull()\n // This loop unrolls recursion in current.resumeWith(param) to
make saner and shorter stack traces on resume\n while (true) {\n with(current) {\n val
completion = resultContinuation\n // Set result and exception fields in the current continuation\n
if (currentException == null) {\n this.result = currentResult\n } else {\n state =
exceptionState\n exception = currentException\n }\n try {\n val
outcome = doResume()\n if (outcome === COROUTINE_SUSPENDED) return\n
currentResult = outcome\n currentException = null\n } catch (exception: dynamic) { // Catch
all exceptions\n currentResult = null\n currentException =
exception.unsafeCast<Throwable>()\n
 }\n releaseIntercepted() // this state machine instance is terminating\n if (completion
is CoroutineImpl) {\n // unrolling recursion via loop\n current = completion\n }
else {\n // top-level completion reached -- invoke and return\n currentException?.let {\n
 completion.resumeWithException(it)\n } ?: completion.resume(currentResult)\n
return\n }\n }\n }\n }\n private fun releaseIntercepted() {\n val intercepted =
intercepted_\n if (intercepted != null && intercepted != this) {\n
context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)\n }\n this.intercepted_ =
CompletedContinuation // just in case\n }\n protected abstract fun doResume(): Any?\n }\n\n
ninternal object
CompletedContinuation
: Continuation<Any?> {\n override val context: CoroutineContext\n get() = error("This continuation is
already complete")\n\n override fun resumeWith(result: Result<Any?>) {\n error("This continuation is
already complete")\n }\n\n override fun toString(): String = "This continuation is already
complete"\n}\n\n
/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:Suppress("UNCHECKED_CAST", "RedundantVisibilityModifier")\n\npackage kotlin\n\nimport
kotlin.contracts.*\nimport kotlin.internal.InlineOnly\nimport kotlin.jvm.JvmField\nimport
kotlin.jvm.JvmInline\nimport kotlin.jvm.JvmName\n\n/**\n * A discriminated union that encapsulates a successful
outcome with a value of type [T]\n * or a failure with an arbitrary [Throwable] exception.\n
*\n\n@SinceKotlin("1.3")\n\n@JvmInline\npublic
value class Result<out T> @PublishedApi internal constructor(\n @PublishedApi\n internal val value: Any?\n)
: Serializable {\n // discovery\n /**\n * Returns `true` if this instance represents a successful outcome.\n *
In this case [isFailure] returns `false`.\n */\n public val isSuccess: Boolean get() = value != null\n /**\n * Returns `true` if this instance represents a failed outcome.\n *
In this case [isSuccess] returns `false`.\n */\n public val isFailure: Boolean get() = value is Failure\n // value & exception retrieval\n /**\n * Returns the
encapsulated value if this instance represents [success][Result.isSuccess] or `null`\n * if it is
[failure][Result.isFailure].\n */\n * This function is a shorthand for `getOrNull { null }` (see [getOrNull]) or\n
 * `fold(onSuccess = { it }, onFailure = { null })` (see [fold]).\n */\n @InlineOnly\n public inline fun
getOrNull(): T? =\n when

```

```

 {
 isFailure -> null
 else -> value as T
 }
 /**
 * Returns the encapsulated
 [Throwable] exception if this instance represents [failure][isFailure] or `null`
 * if it is [success][isSuccess].
 *
 * This function is a shorthand for `fold(onSuccess = { null }, onFailure = { it })` (see [fold]).
 */
 public fun exceptionOrNull(): Throwable? =
 when (value) {
 is Failure -> value.exception
 else ->
 null
 }
 /**
 * Returns a string `Success(v)` if this instance represents [success][Result.isSuccess]
 * where `v` is a string representation of the value or a string `Failure(x)` if
 * it is [failure][isFailure] where `x` is a string representation of the exception.
 */
 public override fun toString(): String =
 when (value) {
 is Failure -> value.toString() // "Failure($exception)"
 else -> "Success($value)"
 }
 // companion with constructors
 /**
 * Companion object for [Result] class that contains its constructor
 functions
 * [success] and [failure].
 */
 public companion object {
 /**
 * Returns an instance
 that encapsulates the given [value] as successful value.
 */
 @Suppress("INAPPLICABLE_JVM_NAME")
 @InlineOnly
 @JvmName("success")
 public inline fun <T> success(value: T): Result<T> =
 Result(value)
 /**
 * Returns an instance that
 encapsulates the given [Throwable] [exception] as failure.
 */
 @Suppress("INAPPLICABLE_JVM_NAME")
 @InlineOnly
 @JvmName("failure")
 public inline fun <T> failure(exception: Throwable): Result<T> =
 Result(createFailure(exception))
 }
 internal class Failure(
 @JvmField
 val exception: Throwable
): Serializable {
 override fun equals(other: Any?): Boolean =
 other is Failure && exception == other.exception
 override fun hashCode(): Int =
 exception.hashCode()
 override fun toString(): String = "Failure($exception)"
 }
 // Creates an instance of internal marker [Result.Failure] class to
 // make sure that this class is not exposed in ABI.
 @PublishedApi
 @SinceKotlin("1.3")
 internal fun createFailure(exception: Throwable): Any =
 Result.Failure(exception)
 /**
 * Throws exception if the result is failure. This internal function minimizes
 * inlined bytecode for [getOrThrow] and makes sure that in the future we can
 * add some exception-augmenting logic here (if needed).
 */
 @PublishedApi
 @SinceKotlin("1.3")
 internal fun Result<*>.throwOnFailure() {
 if (value is Result.Failure) throw value.exception
 }
 /**
 * Calls the specified function [block] and returns its encapsulated result if invocation was successful,
 * catching any [Throwable] exception that was thrown from the [block] function execution and encapsulating it as a failure.
 */
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <R> runCatching(block: () -> R): Result<R> {
 return try {
 Result.success(block())
 } catch (e: Throwable) {
 Result.failure(e)
 }
 }
 /**
 * Calls the specified function [block] with `this` value as its receiver and returns its encapsulated result if invocation was successful,
 * catching any [Throwable] exception that was thrown from the [block] function execution and encapsulating it as a failure.
 */
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {
 return try {
 Result.success(block())
 } catch (e: Throwable) {
 Result.failure(e)
 }
 }
 // -- extensions ---
 /**
 * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or throws the encapsulated [Throwable] exception
 * if it is [failure][Result.isFailure].
 *
 * This function is a shorthand for `getOrElse { throw it }` (see [getOrElse]).
 */
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <T> Result<T>.getOrThrow(): T {
 return value as T
 }
 /**
 * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or the
 * result of [onFailure] function for the encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 *
 * Note, that this function rethrows any [Throwable] exception thrown by [onFailure] function.
 *
 * This function is a shorthand for `fold(onSuccess = { it }, onFailure = onFailure)` (see [fold]).
 */
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <R, T : R> Result<T>.getOrElse(onFailure: (exception: Throwable) -> R): R {
 contract {
 callsInPlace(onFailure,
 InvocationKind.AT_MOST_ONCE)
 }
 return when (val exception = exceptionOrNull()) {
 null ->
 value as T
 else ->
 onFailure(exception)
 }
 }

```

`} }` Returns the encapsulated value if this instance represents `[success][Result.isSuccess]` or the `defaultValue` if it is `[failure][Result.isFailure]`.

`.getOrElse { defaultValue }` (see `[getOrElse]`).

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T : R> Result<T>.getOrElse(defaultValue: R): R` { `if (isFailure) return defaultValue` `return value as T` }

`.onSuccess` Returns the result of `[onSuccess]` for the encapsulated value if this instance represents `[success][Result.isSuccess]` or the result of `[onFailure]` function for the encapsulated `[Throwable]` exception if it is `[failure][Result.isFailure]`.

Note, that this function rethrows any `[Throwable]` exception thrown by `[onSuccess]` or by `[onFailure]` function.

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T> Result<T>.fold( onSuccess: (value: T) -> R, onFailure: (exception: Throwable) -> R): R` { `contract { callsInPlace(onSuccess, InvocationKind.AT_MOST_ONCE)` `callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE)` } `return when (val exception = exceptionOrNull()) { null -> onSuccess(value as T) else -> onFailure(exception) }` }

`.onFailure` Returns the encapsulated result of the given `[transform]` function applied to the encapsulated value if this instance represents `[success][Result.isSuccess]` or the original encapsulated `[Throwable]` exception if it is `[failure][Result.isFailure]`.

Note, that this function rethrows any `[Throwable]` exception thrown by `[transform]` function.

See `[mapCatching]` for an alternative that encapsulates exceptions.

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T> Result<T>.map(transform: (value: T) -> R): Result<R>` { `contract { callsInPlace(transform, InvocationKind.AT_MOST_ONCE)` } `return when { isSuccess -> Result.success(transform(value as T)) else -> Result(value) }` }

`.map` Returns the encapsulated result of the given `[transform]` function applied to the encapsulated value if this instance represents `[success][Result.isSuccess]` or the original encapsulated `[Throwable]` exception if it is `[failure][Result.isFailure]`.

This function catches any `[Throwable]` exception thrown by `[transform]` function and encapsulates it as a failure.

See `[map]` for an alternative that rethrows exceptions from ``transform`` function.

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R>` { `return when { isSuccess -> runCatching { transform(value as T) } else -> Result(value) }` }

`.mapCatching` Returns the encapsulated result of the given `[transform]` function applied to the encapsulated `[Throwable]` exception if this instance represents `[failure][Result.isFailure]` or the original encapsulated value if it is `[success][Result.isSuccess]`.

Note, that this function rethrows any `[Throwable]` exception thrown by `[transform]` function.

See `[recoverCatching]` for an alternative that encapsulates exceptions.

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R>` { `contract { callsInPlace(transform, InvocationKind.AT_MOST_ONCE)` } `return when (val exception = exceptionOrNull()) { null -> this else -> Result.success(transform(exception)) }` }

`.recover` Returns the encapsulated result of the given `[transform]` function applied to the encapsulated `[Throwable]` exception if this instance represents `[failure][Result.isFailure]` or the original encapsulated value if it is `[success][Result.isSuccess]`.

This function catches any `[Throwable]` exception thrown by `[transform]` function and encapsulates it as a failure.

See `[recover]` for an alternative that rethrows exceptions.

`@InlineOnly @SinceKotlin("1.3") public inline fun <R, T : R> Result<T>.recoverCatching(transform: (exception: Throwable) -> R): Result<R>` { `return when (val exception = exceptionOrNull()) { null -> this else -> runCatching { transform(exception) } }` }

`.peek` Performs the given `[action]` on the encapsulated `[Throwable]` exception if this instance represents `[failure][Result.isFailure]`.

Returns the original ``Result`` unchanged.

`@InlineOnly @SinceKotlin("1.3") public inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T>` { `contract { callsInPlace(action, InvocationKind.AT_MOST_ONCE)` } `exceptionOrNull()?.let { action(it) }` `return this` }

`.onFailure` Performs the given `[action]` on the encapsulated value if this instance represents `[success][Result.isSuccess]`.

Returns the original ``Result`` unchanged.

```

*^@InlineOnly^@SinceKotlin("1.3")^public inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit):
Result<T> {^n contract {^n callsInPlace(action, InvocationKind.AT_MOST_ONCE)^n }^n if (isSuccess)
action(value as T)^n return this^}^n// -----^n"/^n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.^n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.^n/^n^package kotlin.coroutines^n^import kotlin.contracts.^n^import
kotlin.coroutines.intrinsics.^n^import kotlin.internal.InlineOnly^}^n/^n * Interface representing a continuation after
a suspension point that returns a value of type `T`.^n *^@SinceKotlin("1.3")^public interface Continuation<in
T> {^n /**^n * The context of the coroutine that corresponds to this continuation.^n *^n public val context:
CoroutineContext^}^n /**^n * Resumes the execution of the
corresponding coroutine passing a successful or failed [result] as the^n * return value of the last suspension
point.^n *^n public fun resumeWith(result: Result<T>)^}^n/^n/**^n * Classes and interfaces marked with this
annotation are restricted when used as receivers for extension^}^n * `suspend` functions. These `suspend` extensions
can only invoke other member or extension `suspend` functions on this particular^}^n * receiver and are restricted
from calling arbitrary suspension functions.^}^n/^n/^n *^@SinceKotlin("1.3")^@Target(AnnotationTarget.CLASS)^@Retention(AnnotationRetention.BINARY)^public
annotation class RestrictsSuspension^}^n/^n/**^n * Resumes the execution of the corresponding coroutine passing
[value] as the return value of the last suspension point.^n *^@SinceKotlin("1.3")^@InlineOnly^public inline
fun <T> Continuation<T>.resume(value: T): Unit =^n resumeWith(Result.success(value))^}^n/^n/**^n * Resumes the
execution of the corresponding coroutine so that the
[exception] is re-thrown right after the^}^n * last suspension point.^}^n/^n/^n *^@SinceKotlin("1.3")^@InlineOnly^public inline fun <T>
Continuation<T>.resumeWithException(exception: Throwable): Unit =^n
resumeWith(Result.failure(exception))^}^n/^n/**^n * Creates a [Continuation] instance with the given [context] and
implementation of [resumeWith] method.^n *^@SinceKotlin("1.3")^@InlineOnly^public inline fun <T>
Continuation(^n context: CoroutineContext,^n crossinline resumeWith: (Result<T>) -> Unit^}^n): Continuation<T>
=^n object : Continuation<T> {^n override val context: CoroutineContext^n get() = context^}^n/^n
override fun resumeWith(result: Result<T>) =^n resumeWith(result)^}^n/^n/**^n * Creates a coroutine
without a receiver and with result type [T].^n * This function creates a new, fresh instance of suspendable
computation every time it is invoked.^n *^}^n * To start executing the created coroutine, invoke `resume(Unit)` on the
returned
[Continuation] instance.^n * The [completion] continuation is invoked when the coroutine completes with a result
or an exception.^n * Subsequent invocation of any resume function on the resulting continuation will produce an
[IllegalStateException].^n *^@SinceKotlin("1.3")^@Suppress("UNCHECKED_CAST")^public fun <T>
(suspend () -> T).createCoroutine(^n completion: Continuation<T>^}^n): Continuation<Unit> =^n
SafeContinuation(createCoroutineUnintercepted(completion).intercepted(), COROUTINE_SUSPENDED)^}^n/^n/**^n
* Creates a coroutine with receiver type [R] and result type [T].^n * This function creates a new, fresh instance of
suspendable computation every time it is invoked.^n *^}^n * To start executing the created coroutine, invoke
`resume(Unit)` on the returned [Continuation] instance.^n * The [completion] continuation is invoked when the
coroutine completes with a result or an exception.^n * Subsequent invocation of any resume function on the resulting
continuation will
produce an [IllegalStateException].^n *^@SinceKotlin("1.3")^@Suppress("UNCHECKED_CAST")^public
fun <R, T> (suspend R.() -> T).createCoroutine(^n receiver: R,^n completion: Continuation<T>^}^n):
Continuation<Unit> =^n SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(),
COROUTINE_SUSPENDED)^}^n/^n/**^n * Starts a coroutine without a receiver and with result type [T].^n * This
function creates and starts a new, fresh instance of suspendable computation every time it is invoked.^n * The
[completion] continuation is invoked when the coroutine completes with a result or an exception.^}^n/^n/^n *^@SinceKotlin("1.3")^@Suppress("UNCHECKED_CAST")^public fun <T> (suspend () ->
T).startCoroutine(^n completion: Continuation<T>^}^n) {^}^n

```

```

createCoroutineUnintercepted(completion).intercepted().resume(Unit)\n}\n\n/**\n * Starts a coroutine with receiver
type [R] and result type [T].\n * This function creates and starts a new, fresh instance of suspendable
computation every time it is invoked.\n * The [completion] continuation is invoked when the coroutine completes
with a result or an exception.\n */\n@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <R,
T> (suspend R.() -> T).startCoroutine(\n receiver: R,\n completion: Continuation<T>)\n {\n
createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)\n}\n\n/**\n * Obtains the current
continuation instance inside suspend functions and suspends\n * the currently running coroutine.\n */\n * In this
function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously
in\n * the same stack-frame where the suspension function is run or asynchronously later in the same thread or\n *
from a different thread of execution. Subsequent invocation of any resume function will produce an
[IllegalStateException].\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic suspend inline fun <T>
suspendCoroutine(crossinline block: (Continuation<T>)\n
-> Unit): T {\n contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n return
suspendCoroutineUninterceptedOrReturn { c: Continuation<T> ->\n val safe =
SafeContinuation(c.intercepted())\n block(safe)\n safe.getOrThrow()\n }\n}\n\n/**\n * Returns the
context of the current coroutine.\n */\n@SinceKotlin("1.3")\n@Suppress("WRONG_MODIFIER_TARGET")\n@InlineOnly\npublic suspend inline
val coroutineContext: CoroutineContext\n get() {\n throw NotImplementedError("Implemented as
intrinsic")\n }\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
/\n\npackage kotlin.coroutines.intrinsics\n\nimport kotlin.coroutines.\nimport kotlin.internal.InlineOnly\n\n/**\n *
Starts an unintercepted coroutine without a receiver and with result type [T] and executes it until its
first suspension.\n * Returns the result of the coroutine or throws its exception if it does not suspend or
[COROUTINE_SUSPENDED] if it suspends.\n * In the latter case, the [completion] continuation is invoked when
the coroutine completes with a result or an exception.\n */\n * The coroutine is started directly in the invoker's thread
without going through the [ContinuationInterceptor] that might\n * be present in the completion's
[CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation\n * context is established.\n
*/\n * This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the
execution of the suspended\n * coroutine using a reference to the suspending function.\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic actual inline fun <T> (suspend () ->
T).startCoroutineUninterceptedOrReturn(\n completion: Continuation<T>)\n): Any? =
this.asDynamic()(completion, false)\n\n/**\n * Starts an unintercepted coroutine
with receiver type [R] and result type [T] and executes it until its first suspension.\n * Returns the result of the
coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n * In the
latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.\n
*/\n * The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor]
that might\n * be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a
proper invocation\n * context is established.\n */\n * This function is designed to be used from inside of
[suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference
to the suspending function.\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic actual inline fun <R, T> (suspend
R.() -> T).startCoroutineUninterceptedOrReturn(\n receiver: R,\n completion: Continuation<T>)\n):
Any? = this.asDynamic()(receiver, completion, false)\n\n@InlineOnly\ninternal actual inline fun <R, P, T>
(suspend R.(P) -> T).startCoroutineUninterceptedOrReturn(\n receiver: R,\n param: P,\n completion:
Continuation<T>)\n): Any? = this.asDynamic()(receiver, param, completion, false)\n\n/**\n * Creates unintercepted
coroutine without receiver and with result type [T].\n * This function creates a new, fresh instance of suspendable
computation every time it is invoked.\n */\n * To start executing the created coroutine, invoke `resume(Unit)` on the
returned [Continuation] instance.\n * The [completion] continuation is invoked when coroutine completes with
result or exception.\n */\n * This function returns unintercepted continuation.\n * Invocation of `resume(Unit)` starts

```



coroutine immediately in the invoker's call stack without going through the `[ContinuationInterceptor]` that might be present in the completion's `[CoroutineContext]`. It is the invoker's responsibility to ensure that a proper invocation context is established. Note that `[completion]` of this function may get invoked in an arbitrary context. `[Continuation.intercepted]` can be used to acquire the intercepted continuation. Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and `[completion]` happens in the invocation context established by `[ContinuationInterceptor]`. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception.

```

@SinceKotlin("1.3")
public actual fun <T>
(suspend () -> T).createCoroutineUnintercepted(
 completion: Continuation<T>): Continuation<Unit> =
 // Kotlin/JS suspend lambdas have an extra parameter `suspended`
 if (this.asDynamic().length == 2) {
 // When `suspended` is true the continuation is created, but not executed
 this.asDynamic()(completion, true)
 } else {
 createCoroutineFromSuspendFunction(completion) {
 this.asDynamic()(completion)
 }
 }
}

```

Creates unintercepted coroutine with receiver type `[R]` and result type `[T]`. This function creates a new, fresh instance of suspendable computation every time it is invoked. To start executing the created coroutine, invoke `resume(Unit)` on the returned `[Continuation]` instance. The `[completion]` continuation is invoked when coroutine completes with result or exception. This function returns unintercepted continuation. Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the `[ContinuationInterceptor]` that might be present in the completion's `[CoroutineContext]`. It is the invoker's responsibility to ensure that a proper invocation context is established. Note that `[completion]` of this function may get invoked in an arbitrary context.

`[Continuation.intercepted]` can be used to acquire the intercepted continuation. Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and `[completion]` happens in the invocation context established by `[ContinuationInterceptor]`. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception.

```

@SinceKotlin("1.3")
public actual fun <R, T> (suspend R.() ->
T).createCoroutineUnintercepted(
 receiver: R,
 completion: Continuation<T>): Continuation<Unit> =
 // Kotlin/JS suspend lambdas have an extra parameter `suspended`
 if (this.asDynamic().length == 3) {
 // When `suspended` is true the continuation is created, but not executed
 this.asDynamic()(receiver, completion, true)
 } else {
 createCoroutineFromSuspendFunction(completion) {
 this.asDynamic()(receiver, completion)
 }
 }
}

```

Intercepts this continuation with `[ContinuationInterceptor]`. This function shall be used on the immediate result of `[createCoroutineUnintercepted]` or `[suspendCoroutineUninterceptedOrReturn]`, in which case it checks for `[ContinuationInterceptor]` in the continuation's `[context][Continuation.context]`, invokes `[ContinuationInterceptor.interceptContinuation]`, caches and returns the result. If this function is invoked on other `[Continuation]` instances it returns this continuation unchanged.

```

@SinceKotlin("1.3")
public actual fun <T> Continuation<T>.intercepted(): Continuation<T> =
 (this as? CoroutineImpl)?.intercepted() ?:
 this
private inline fun <T> createCoroutineFromSuspendFunction(
 completion: Continuation<T>,
 crossinline block: () -> Any?): Continuation<Unit> {
 @Suppress("UNCHECKED_CAST")
 return object
 : CoroutineImpl(completion as Continuation<Any?>) {
 override fun doResume(): Any? {
 exception?.let { throw it }
 return block()
 }
 }
}

```

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

package kotlin.js  
Mirrors signature from JS IR BE  
Used for  
js.translator/testData/box/number/mulInt32.kt  
@library  
@jsName("imulEmulated")  
@Suppress("UNUSED\_PARAMETER")  
internal fun imul(x: Int, y: Int): Int =  
definedExternally  
@Suppress("NOTHING\_TO\_INLINE")  
internal inline fun isArrayish(o: dynamic) =  
js("Kotlin").isArrayish(o)  
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the



```

\n\n@file:Suppress("UNUSED_PARAMETER")\n\npackage kotlin.js\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun jsDeleteProperty(obj: Any, property: Any) {\n js("delete obj[property]")\n}\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =\n js("lhs | rhs").unsafeCast<Int>(), /\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n *\n *\n Returns this value with the sign bit same as of the [sign] value.\n *\n *\n If [sign] is `NaN` the sign of the result is undefined.\n\n*\n\n@SinceKotlin("1.2")\n\npublic actual fun Double.withSign(sign: Double): Double {\n val thisSignBit = js("Kotlin").doubleSignBit(this).unsafeCast<Int>()\n val newSignBit = js("Kotlin").doubleSignBit(sign).unsafeCast<Int>()\n return if (thisSignBit == newSignBit) this else -this\n}, /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n *\n *\n Returns a bit representation of the specified floating-point value as [Long]\n *\n according to the IEEE 754 floating-point "double format" bit layout.\n\n*\n\n@SinceKotlin("1.2")\n\n@library("doubleToBits")\n\npublic actual fun Double.toBits(): Long = definedExternally\n\n *\n *\n Returns a bit representation of the specified floating-point value as [Long]\n *\n according to the IEEE 754 floating-point "double format" bit layout,\n *\n preserving `NaN` values exact layout.\n\n*\n\n@SinceKotlin("1.2")\n\n@library("doubleToRawBits")\n\npublic actual fun Double.toRawBits(): Long = definedExternally\n\n *\n *\n Returns the [Double] value corresponding to a given bit representation.\n\n*\n\n@SinceKotlin("1.2")\n\n@kotlin.internal.InlineOnly\n\npublic actual inline fun Double.Companion.fromBits(bits: Long): Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()\n\n *\n *\n Returns a bit representation of the specified floating-point value as [Int]\n *\n according to the IEEE 754 floating-point "single format" bit layout.\n *\n *\n Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\n *\n so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.\n\n*\n\n@SinceKotlin("1.2")\n\n@library("floatToBits")\n\npublic actual fun Float.toBits(): Int = definedExternally\n\n *\n *\n Returns a bit representation of the specified floating-point value as [Int]\n *\n according to the IEEE 754 floating-point "single format" bit layout,\n *\n preserving `NaN` values exact layout.\n *\n *\n Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\n *\n so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.\n\n*\n\n@SinceKotlin("1.2")\n\n@library("floatToRawBits")\n\npublic actual fun Float.toRawBits(): Int = definedExternally\n\n *\n *\n Returns the [Float] value corresponding to a given bit representation.\n\n*\n\n@SinceKotlin("1.2")\n\n@kotlin.internal.InlineOnly\n\npublic actual inline fun Float.Companion.fromBits(bits: Int): Float = js("Kotlin").floatFromBits(bits).unsafeCast<Float>()\n\n *\n *\n Suppress("NOTHING_TO_INLINE")\n\n *\n *\n internal inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()\n\n *\n *\n internal inline val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()\n\n *\n *\n internal inline val Long.high: Int get() = this.asDynamic().getHighBits().unsafeCast<Int>()\n\n */\n\n *\n *\n Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *\n *\n Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n *\n *\n import kotlin.reflect.KClass\n\n *\n *\n @PublishedApi\n\n *\n *\n internal fun <T : Annotation> KClass<*>.findAssociatedObject(@Suppress("UNUSED_PARAMETER") annotationClass: KClass<T>): Any? {\n {\n // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir.\n return null\n }\n }, /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n *\n *\n Returns a string representation of this [Long] value in the specified [radix].\n\n *\n *\n @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n\n}

```

```

*\n@SinceKotlin("1.2")\npublic actual fun Long.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n@PublishedApi\n@Suppress("NOTHING_TO_INLINE")\n@JsPolyfill("\n\n"nif (typeof
Array.prototype.fill
=== "undefined") {\n // Polyfill from https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n Object.defineProperty(Array.prototype,
'fill', {\n value: function (value) {\n // Steps 1-2.\n if (this == null) {\n throw new
TypeError('this is null or not defined');\n }\n\n var O = Object(this);\n\n // Steps 3-5.\n
var len = O.length >>> 0;\n\n // Steps 6-7.\n var start = arguments[1];\n var relativeStart = start
>> 0;\n\n // Step 8.\n var k = relativeStart < 0 ?\n Math.max(len + relativeStart, 0) :\n
Math.min(relativeStart, len);\n\n // Steps 9-10.\n var end = arguments[2];\n var
relativeEnd = end === undefined ?\n len : end >> 0;\n\n // Step 11.\n var finalValue
= relativeEnd < 0 ?\n Math.max(len + relativeEnd, 0) :\n Math.min(relativeEnd,
len);\n\n // Step 12.\n while (k < finalValue) {\n O[k] = value;\n k++;\n }\n\n
// Step 13.\n return O;\n }\n });\n\n[Int8Array, Int16Array, Uint16Array, Int32Array,
Float32Array, Float64Array].forEach(function (TypedArray) {\n if (typeof TypedArray.prototype.fill ===
"undefined") {\n Object.defineProperty(TypedArray.prototype, 'fill', {\n value: Array.prototype.fill\n
});\n }\n })\n\n"")\n\ninternal inline fun Any.nativeFill(element: Any?, fromIndex: Int, toIndex: Int): Unit {\n
asDynamic().fill(element, fromIndex, toIndex)\n}\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt
file.\n */\n\npackage
kotlin.js\n\n@PublishedApi\n@Suppress("NOTHING_TO_INLINE")\n@JsPolyfill("\n\n"n[Int8Array,
Int16Array, Uint16Array, Int32Array, Float32Array, Float64Array].forEach(function (TypedArray) {\n if (typeof
TypedArray.prototype.sort === "undefined") {\n Object.defineProperty(TypedArray.prototype, 'sort', {\n
value: function(compareFunction) {\n compareFunction = compareFunction || function (a, b) {\n
if (a < b) return -1;\n if (a > b) return 1;\n if (a === b) {\n if (a !== 0) return
0;\n var ia = 1 / a;\n return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n }\n
return a !== a ? (b !== b ? 0 : 1) : -1\n }\n return Array.prototype.sort.call(this, compareFunction
|| totalOrderComparator);\n }\n });\n }\n })\n\n"")\n\ninternal
inline fun Any.nativeSort(noinline comparison: (a: dynamic, b: dynamic) -> Int = js("undefined")): Unit {\n
asDynamic().sort(comparison)\n}\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\n// 1343 ranges
totally\n\nprivate object Category {\n val decodedRangeStart: IntArray\n val decodedRangeCategory: IntArray\n
\n init {\n val toBase64 =
"\n\n"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+\n\n val fromBase64 =
IntArray(128)\n for (i in toBase64.indices) {\n fromBase64[toBase64[i].code] = i\n }\n\n //
rangeStartDiff.length = 1482\n val rangeStartDiff
=
"\n\n"gBCFEDCKCDCaDDaDBhBCEEDDDDDDEDXBHYBH5BRwBGDCHDCIDFHDCHFDCDEIRTEE7BGHDDJI
CBbSEMOfGERwDEDDDDDECEFCRBjBFDCYFFCCzBvBjBBFC3BOhDBmBDGpBDDCtBBJlBEECLGDFC
LDCgBBKVKEDiDDHCFECECKCEODBebC5CLBOKhBJDDDDWEBHFCCPZDEL1BVBSLPgBB2BDB
DICFBHKCKCPDBHEDWBHEDDDDEDEDIBDGDCKCCGDDDCGECWBFMDDCEDDDCHDDHKDDBK
DBHFCEWBFGBDDDFEDBPDDKCHBGDCHEDWBFGBDCEDEDBHDDGDCKCGJEGDBFDDFDDEDDDDME

```



```

Int {
 return when {
 code < 0x20 -> code
 code < 0x400 -> if ((ch and 1) == 1) code shr 5 else code
 and 0x1f
 else -> when (ch % 3) {
 2 -> code shr 10
 1 -> (code shr 5) and 0x1f
 else -> code and 0x1f
 }
 }
}

/**
 * Returns the Unicode general category of this character
 * as an Int.
 */
internal fun Char.getCategoryValue(): Int {
 val ch = this.code
 val index =
 binarySearchRange(Category.decodedRangeStart, ch)
 val start = Category.decodedRangeStart[index]
 val code = Category.decodedRangeCategory[index]
 val value = categoryValueFrom(code, ch - start)
 return if (value == 17) CharCategory.UNASSIGNED.value else value
}

internal fun decodeVarLenBase64(base64:
String, fromBase64: IntArray, resultLength: Int): IntArray {
 val result = IntArray(resultLength)
 var index = 0
 var int = 0
 var shift = 0
 for (char in base64) {
 val sixBit = fromBase64[char.code]
 int = int or ((sixBit and 0x1f) shl shift)
 if (sixBit < 0x20) {
 result[index++] = int
 int = 0
 shift = 0
 } else {
 shift += 5
 }
 }
 return result
}

/**
 * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

import
kotlin.js.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Reverses elements in the list in-
 * place.
 */
public actual fun <T> MutableList<T>.reverse(): Unit {
 val midPoint = (size / 2) - 1
 if (midPoint < 0) return
 var reverseIndex = lastIndex
 for (index in 0..midPoint) {
 val tmp = this[index]
 this[index] = this[reverseIndex]
 this[reverseIndex] = tmp
 reverseIndex--
 }
}

/**
 * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */
package kotlin.text

// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

// 37 ranges
totally
private object Digit {
 internal val rangeStart = arrayOf(
 0x0030, 0x0660, 0x06f0, 0x07c0,
 0x0966, 0x09e6, 0x0a66, 0x0ae6, 0x0b66, 0x0be6, 0x0c66, 0x0ce6, 0x0d66, 0x0de6, 0x0e50, 0x0ed0, 0x0f20,
 0x1040, 0x1090, 0x17e0,
 0x1810, 0x1946, 0x19d0, 0x1a80, 0x1a90, 0x1b50, 0x1bb0, 0x1c40, 0x1c50,
 0xa620, 0xa8d0, 0xa900, 0xa9d0, 0xa9f0, 0xaa50, 0xabf0, 0xff10,
)
}

/**
 * Returns the index of the
 * largest element in [array] smaller or equal to the specified [needle],
 * or -1 if [needle] is smaller than the smallest
 * element in [array].
 */
internal fun binarySearchRange(array: IntArray, needle: Int): Int {
 var bottom = 0
 var top = array.size - 1
 var middle = -1
 var value = 0
 while (bottom <= top) {
 middle = (bottom +
 top) / 2
 value = array[middle]
 if (needle > value)
 bottom = middle + 1
 else if (needle ==
 value)
 return middle
 else
 top = middle - 1
 }
 return middle - (if (needle < value) 1 else 0)
}

/**
 * Returns an integer from 0..9 indicating the digit this character represents,
 * or -1 if this character
 * is not a digit.
 */
internal fun Char.digitToIntImpl(): Int {
 val ch = this.code
 val index =
 binarySearchRange(Digit.rangeStart, ch)
 val diff = ch - Digit.rangeStart[index]
 return if (diff < 10) diff else -
 1
}

/**
 * Returns
 * `true` if this character is a digit.
 */
internal fun Char.isDigitImpl(): Boolean {
 return digitToIntImpl() >=
 0
}

/**
 * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.text

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

// 222 ranges
totally
private object Letter {
 val decodedRangeStart: IntArray
 val decodedRangeLength: IntArray
 val decodedRangeCategory:
 IntArray
 init {
 val toBase64 =
 "\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
 val fromBase64 =
 IntArray(128)
 for (i in toBase64.indices) {
 fromBase64[toBase64[i].code] = i
 }
 rangeStartDiff.length = 356
 val rangeStartDiff
 =
 "\"hCgBpCQGYHZH5BRpBPPPPPRMP5BPPICPP6BkEPPPPcXPzBvBrB3BOiDoBHwD+E3DauCnFmBmB2D

```

```

6E1BIBTiBmBIBP5BhBiBrBvBjBqBnBPRtBiCmCtBIB0BmB5BiB7BmBgEmChBZgCoEoGVpBsFrhBPqKQ2B
wBYoFgB4CJuTiEvBuCuDrF5DgEgFIJ1DgFmBQtBsBRGsB+BPiBID1EIjDPRPPPPPPPPGQSQS/DxENVNU+
B9zCwBwBPPCkDPNnBPqDYY1R8B7FkFgTgwGgwUwmBgKwBuBScmEP/BPPPPPrBP8B7F1B/ErBqC6B7B
iBmBfQsBUwCw/KwqIwLwETPcPjQgJxFgBIBsD"\n val diff = decodeVarLenBase64(rangeStartDiff,
fromBase64, 222)\n val start = IntArray(diff.size)\n for (i in diff.indices) {\n if (i == 0) start[i] =
diff[i]\n else start[i] = start[i - 1] + diff[i]\n }\n decodedRangeStart = start\n \n //
rangeLength.length = 328\n val rangeLength =
\"aaMBXHYH5BRpBPPPPPRMP5BPPICPPzBDOOPPcXPzBvBjB3BOhDmBBpB7DoDYxB+EiBP1DoExBkB
QhBekBPmBgBhBctBiBMWOOXhCsBpBkBUV3Ba4BkB0DiCgBXgBtD4FSdBfPhBPpKP0BvBXjEQ2CGsT8Dh
BtCqDpFvD1D3E0lrD2EkBjRBDObSb+BPiBIB1EIjDPPPPPPPPPPGPPMNLsBNPNPKcVbBPPCkDPBmBPh
DXXgD4B6FzEgDguG9vUtkB9JcuBSckEP/BPPPPPPBPf4FrBjEhBpC3B5BKaWPrBOWck/KsCuLqDHPbPxPsFt
EaaqDL\"\n
 decodedRangeLength = decodeVarLenBase64(rangeLength, fromBase64, 222)\n \n //
rangeCategory.length = 959\n val rangeCategory =
\"GFjgggUHGFFZZZmzpz5qB6s6020B60ptltB6smt2sB60mz22B1+vv+8BZZ5s2850BW5q1ymtB506smzBF3q1
q1qB1q1q1+Bgii4wDTm74g3KigxqM60q1q1Bq1o1q1BF1qlrqrBZ2q5wprBGFZWWZGHFsjiioLowgmOowjkw
CkgoiIk7ligGogioBkwwiYkzj2oNoi+sbkwj04DghhkQ8wgiYkgoioDsgnkwC4gikQ//v+85BkwvoIsgoyl4ygu0whiw
Eowri4CoghsJowgqYowgm4DkwgsY/nwnzPowhmYkg6wI8yggZswikwHgxgmIoxgqYkwgk4DkxgmIkgoioBsgsso
BgzgyI8g9gL8g9kI0wgwJoxgkoC0wgioFkw/wI0w53iF4gioYowjmgBHGq1qkgwBF1q1q8qBHwghuIwghyKk0go
QkwgoQk3goQHGFHkyg0pBgxj6IoinkxDswno7Ikwhz9Bo0gioB8z48Rwli0xN0mpjoX8w78pDwltoqKHFGGwwg
sIHfH3q1q16BFHWFZ1q10q1B2qlwq1B1q10q1B2q1yq1B6q1gq1Biq1qhxBir1qp1Bqt1q1qB1g1q1+B//3q16B///q
1qBH/qlq9Bholq9B1i00a1q10qD1op1HkwmigEigiy6Cptogq1Bixo1kDq7/j00B2qgoBwGFm1lz50B6s5q1+BG
WhggzhwBFFhgk4//Bo2jigE8wguI8wguI8wguUog1qoB4qjmIwwi2KkgYHHH4IBgiFWkgIWoghssMmz5smrBZ
3q1y50B5sm7gzBtz1smzB5smz50BqzqtmzB5sgzqzBF2/9//5BowgoIwmnkzPkwgk4C8ys65BkgoqI0wgy6FghquZo
2giY0ghiIsgH24B4ghsQ8QF/v1q1OFs0O8iCHHF1qggz/B8wg6Iznv+//B08QgohsjK0QGfK7hsQ4gB\"\n
 decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 222)\n }\n}\n\n/*\n * Returns
`true` if this character is a letter.\n */\ninternal fun Char.isLetterImpl(): Boolean {\n return getLetterType() !=
0\n}\n}\n\n/*\n * Returns `true` if this character is a lower case letter, or it has contributory property
`Other_Lowercase`.\n */\ninternal fun Char.isLowerCaseImpl(): Boolean {\n return getLetterType() == 1 ||
code.isOtherLowercase()\n}\n}\n\n/*\n * Returns `true` if this character is an upper case letter, or it has contributory
property `Other_Uppercase`.\n */\ninternal fun Char.isUpperCaseImpl(): Boolean {\n return getLetterType() == 2
|| code.isOtherUppercase()\n}\n}\n\n/*\n * Returns\n * - `1` if the character is a lower case letter,\n * - `2` if the
character is an
upper case letter,\n * - `3` if the character is a letter but not a lower or upper case letter,\n * - `0` otherwise.\n
*/\nprivate fun Char.getLetterType(): Int {\n val ch = this.code\n val index =
binarySearchRange(Letter.decodedRangeStart, ch)\n val rangeStart = Letter.decodedRangeStart[index]\n val
rangeEnd = rangeStart + Letter.decodedRangeLength[index] - 1\n val code =
Letter.decodedRangeCategory[index]\n if (ch > rangeEnd) {\n return 0\n }\n val lastTwoBits = code
and 0x3\n if (lastTwoBits == 0) { // gap pattern\n var shift = 2\n var threshold = rangeStart\n for (i
in 0..1) {\n threshold += (code shr shift) and 0x7f\n if (threshold > ch) {\n return 3\n
}\n shift += 7\n threshold += (code shr shift) and 0x7f\n if (threshold > ch) {\n return
0\n }\n shift += 7\n }\n return 3\n }\n if (code <= 0x7) {\n return lastTwoBits\n }\n val distance = (ch - rangeStart)\n val shift = if (code <=
0x1F) distance % 2 else distance\n return (code shr (2 * shift)) and 0x3\n}\n}\n\n\"/>\n\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n\nNOTE:
THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nprivate object OtherLowercase {\n internal

```





```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.elementAt(index: Int):
UShort {\n return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")}
}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.asList(): List<UInt> {\n
return object : AbstractList<UInt>(), RandomAccess {\n override val size: Int get() = this@asList.size\n
override fun
isEmpty(): Boolean = this@asList.isEmpty()\n override fun contains(element: UInt): Boolean =
this@asList.contains(element)\n override fun get(index: Int): UInt {\n
AbstractList.checkElementIndex(index, size)\n return this@asList[index]\n }\n override fun
indexOf(element: UInt): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UInt)
return -1\n return this@asList.indexOf(element)\n }\n override fun lastIndexOf(element: UInt): Int
{\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UInt) return -1\n return
this@asList.lastIndexOf(element)\n }\n}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.asList(): List<ULong>
{\n return object : AbstractList<ULong>(), RandomAccess {\n override val size: Int get() = this@asList.size\n
override fun isEmpty(): Boolean = this@asList.isEmpty()\n override fun contains(element: ULong):
Boolean = this@asList.contains(element)\n override fun get(index: Int): ULong {\n
AbstractList.checkElementIndex(index, size)\n return this@asList[index]\n }\n override fun
indexOf(element: ULong): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is
ULong) return -1\n return this@asList.indexOf(element)\n }\n override fun lastIndexOf(element:
ULong): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is ULong) return -1\n
return this@asList.lastIndexOf(element)\n }\n }\n}\n}\n\n/**\n * Returns a [List] that wraps the original
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.asList():
List<UByte> {\n return object : AbstractList<UByte>(), RandomAccess {\n override val size:
Int get() = this@asList.size\n override fun isEmpty(): Boolean = this@asList.isEmpty()\n override fun
contains(element: UByte): Boolean = this@asList.contains(element)\n override fun get(index: Int): UByte {\n
AbstractList.checkElementIndex(index, size)\n return this@asList[index]\n }\n override fun
indexOf(element: UByte): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is
UByte) return -1\n return this@asList.indexOf(element)\n }\n override fun lastIndexOf(element:
UByte): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UByte) return -1\n
return this@asList.lastIndexOf(element)\n }\n }\n}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.asList(): List<UShort>
{\n return object : AbstractList<UShort>(), RandomAccess
{\n override val size: Int get() = this@asList.size\n override fun isEmpty(): Boolean =
this@asList.isEmpty()\n override fun contains(element: UShort): Boolean = this@asList.contains(element)\n
override fun get(index: Int): UShort {\n AbstractList.checkElementIndex(index, size)\n return
this@asList[index]\n }\n override fun indexOf(element: UShort): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UShort) return -1\n return
this@asList.indexOf(element)\n }\n override fun lastIndexOf(element: UShort): Int {\n @Suppress("USELESS_CAST")\n if ((element as Any?) !is UShort) return -1\n return
this@asList.lastIndexOf(element)\n }\n }\n }\n}\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the
license/LICENSE.txt file.\n
*/\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\n9 ranges
totally\n\n/**\n * Returns `true` if this character is a whitespace.\n
*\n@internal fun Char.isWhitespaceImpl(): Boolean
{\n val ch = this.code\n return ch in 0x0009..0x000d\n || ch in 0x001c..0x0020\n || ch == 0x00a0\n
|| ch > 0x1000 && (\n ch == 0x1680\n || ch in 0x2000..0x200a\n || ch == 0x2028\n

```

```

 || ch == 0x2029\n || ch == 0x202f\n || ch == 0x205f\n || ch == 0x3000\n
)\n}\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin\n\n\npublic
actual fun interface Comparator<T> {\n @JsName("compare")\n public actual fun compare(a: T, b: T):
Int\n}\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.js\n\nimport
kotlin.annotation.AnnotationTarget.*\n\n@Target(FUNCTION)\n@Deprecated("Use inline extension function with
body using dynamic")\npublic annotation class nativeGetter\n\n@Target(FUNCTION)\n@Deprecated("Use inline
extension function with body using dynamic")\npublic annotation class
nativeSetter\n\n@Target(FUNCTION)\n@Deprecated("Use inline extension function with body using
dynamic")\npublic annotation class nativeInvoke\n\n@Target(CLASS, FUNCTION, PROPERTY)\n\ninternal
annotation class library(public val name: String = "")\n\n@Target(CLASS)\n\ninternal annotation class
marker\n\n/*\n * Gives a declaration (a function, a property
or a class) specific name in JavaScript.\n * This may be useful in the following cases:\n * * There are two
functions for which the compiler gives same name in JavaScript, you can\n * mark one with `@JsName(...)` to
prevent the compiler from reporting error.\n * * You are writing a JavaScript library in Kotlin. The compiler
produces mangled names\n * for functions with parameters, which is unnatural for usual JavaScript developer.\n *
You can put `@JsName(...)` on functions you want to be available from JavaScript.\n * * For some reason you
want to rename declaration, e.g. there's common term in JavaScript\n * for a concept provided by the declaration,
which is uncommon in Kotlin.\n * Example:\n * ```\n * kotlin\n * class Person(val name: String) {\n * fun
hello() {\n * println("Hello $name!")\n * }\n * @JsName("helloWithGreeting")\n * fun
hello(greeting: String) {\n * println("$greeting $name!")\n * }\n * }\n * ```\n * @property name the name which compiler uses both for declaration itself and for all references to
the declaration.\n * It's required to denote a valid JavaScript identifier.\n *
*/\n\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR,
PROPERTY_GETTER, PROPERTY_SETTER)\n\npublic actual annotation class JsName(actual val name:
String)\n\n/*\n * Denotes an `external` declaration that must be imported from native JavaScript library.\n *
The compiler produces the code relevant for the target module system, for example, in case of CommonJS,\n * it
will import the declaration via the `require(...)` function.\n * The annotation can be used on top-level external
declarations (classes, properties, functions) and files.\n * In case of file (which can't be `external`) the following rule
applies: all the declarations in\n * the file must be `external`. By applying `@JsModule(...)` on a file you tell the
compiler to import a
JavaScript object\n * that contain all the declarations from the file.\n * Example:\n * ```\n * kotlin\n *
@JsModule("jquery")\n * external abstract class JQuery() {\n * // some declarations here\n * }\n *
@JsModule("jquery")\n * external fun JQuery(element: Element): JQuery\n * ```\n * @property import name
of a module to import declaration from.\n * It is not interpreted by the Kotlin compiler, it's passed as is
directly to the target module system.\n * @see JsNonModule\n
*/\n\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\n\npublic
annotation class JsModule(val import: String)\n\n/*\n * Denotes an `external` declaration that can be used without
module system.\n * By default, an `external` declaration is available regardless your target module system.\n *
However, by applying [JsModule] annotation you can make a declaration unavailable to *plain* module system.\n *
Some JavaScript libraries are distributed both as
a standalone downloadable piece of JavaScript and as a module available\n * as an npm package.\n * To tell the
Kotlin compiler to accept both cases, you can augment [JsModule] with the `@JsNonModule` annotation.\n *
For example:\n * ```\n * kotlin\n * @JsModule("jquery")\n * @JsNonModule\n * @JsName("$")\n * external
abstract class JQuery() {\n * // some declarations here\n * }\n * @JsModule("jquery")\n * @JsNonModule\n

```

```

* @JsName("$")\n * external fun JQuery(element: Element): JQuery\n * ```\n * @see JsModule\n
*\n @Retention(AnnotationRetention.BINARY)\n @Target(CLASS, PROPERTY, FUNCTION, FILE)\n public
annotation class JsNonModule\n\n /**\n * Adds prefix to `external` declarations in a source file.\n * JavaScript
does not have concept of packages (namespaces). They are usually emulated by nested objects.\n * The compiler
turns references to `external` declarations either to plain unprefixed names (in case of *plain* modules)\n * or to
plain imports.\n * However,
if a JavaScript library provides its declarations in packages, you won't be satisfied with this.\n * You can tell the
compiler to generate additional prefix before references to `external` declarations using the `@JsQualifier(...)`\n *
annotation.\n * Note that a file marked with the `@JsQualifier(...)` annotation can't contain non-`external`
declarations.\n * Example:\n * ```\n * @file:JsQualifier("my.jsPackageName")\n * package
some.kotlinPackage\n * external fun foo(x: Int)\n * external fun bar(): String\n * ```\n * @property value
the qualifier to add to the declarations in the generated code.\n * It must be a sequence of valid JavaScript
identifiers separated by the `.` character.\n * Examples of valid qualifiers are: `foo`, `bar.Baz`, `_$0.f`.\n *
*\n * @see JsModule\n *\n @Retention(AnnotationRetention.BINARY)\n @Target(AnnotationTarget.FILE)\n public
annotation class JsQualifier(val value: String)\n\n /**\n * Exports top-level declaration
on JS platform.\n * Compiled module exposes declarations that are marked with this annotation without name
mangling.\n * This annotation can be applied to either files or top-level declarations.\n * It is currently
prohibited to export the following kinds of declarations:\n * * `expect` declarations\n * * inline functions with
reified type parameters\n * * suspend functions\n * * secondary constructors without `@JsName`\n * *
extension properties\n * * enum classes\n * * annotation classes\n * Signatures of exported declarations must
only contain "exportable" types:\n * * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`,
`Double`\n * * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * *
`Array<exportable-type>`\n * * Function types with exportable parameters and return types\n * * `external` or
`@JsExport` classes and interfaces\n * * Nullable counterparts of types above\n
* * Unit return type. Must not be nullable\n * This annotation is experimental, meaning that restrictions
mentioned above are subject to change.\n
*\n @ExperimentalJsExport\n @Retention(AnnotationRetention.BINARY)\n @Target(CLASS, PROPERTY,
FUNCTION, FILE)\n @SinceKotlin("1.3")\n public actual annotation class JsExport\n\n /**\n * Forces a top-level
property to be initialized eagerly, opposed to lazily on the first access to file and/or property.\n
*\n @ExperimentalStdlibApi\n @Retention(AnnotationRetention.BINARY)\n @Target(AnnotationTarget.PROPER
TY)\n @SinceKotlin("1.6")\n @Deprecated("This annotation is a temporal migration assistance and may be
removed in the future releases, please consider filing an issue about the case where it is needed")\n public annotation
class EagerInitialization", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt
file.\n *\n package kotlin.jvm\n\n // these are used in common generated code in stdlib\n\n // TODO: find how to
deprecate these ones\n\n @Target(AnnotationTarget.FIELD)\n @Retention(AnnotationRetention.SOURCE)\n public
actual annotation class Volatile\n\n @Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY_GETTER,
AnnotationTarget.PROPERTY_SETTER)\n @Retention(AnnotationRetention.SOURCE)\n public actual annotation
class Synchronized\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n package kotlin.collections\n\n /**\n * Provides a skeletal implementation of the
[MutableCollection] interface.\n * @param E the type of elements contained in the collection. The collection is
invariant in its element type.\n *\n public actual abstract class AbstractMutableCollection<E> protected actual
constructor() : AbstractCollection<E>(),
MutableCollection<E> {\n\n actual abstract override fun add(element: E): Boolean\n\n actual override fun
remove(element: E): Boolean {\n\n checkIsMutable()\n\n val iterator = iterator()\n\n while

```

```

(iterator.hasNext()) {\n if (iterator.next() == element) {\n iterator.remove()\n return true\n }\n }\n return false\n }\n\n actual override fun addAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n var modified = false\n for (element in elements) {\n if (add(element)) modified = true\n }\n return modified\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return (this as MutableIterable<E>).removeAll { it in elements }\n }\n\n actual override fun retainAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return (this as MutableIterable<E>).removeAll\n {\n !it in elements\n }\n }\n\n actual override fun clear(): Unit {\n checkIsMutable()\n val iterator = this.iterator()\n while (iterator.hasNext()) {\n iterator.next()\n iterator.remove()\n }\n }\n\n @Deprecated("Provided so that subclasses inherit this function", level = DeprecationLevel.HIDDEN)\n @JsName("toJSON")\n protected fun toJSON(): Any = this.toArray()\n\n /**\n * This method is called every time when a mutating method is called on this mutable collection.\n * Mutable collections that are built (frozen) must throw `UnsupportedOperationException`.\n */\n internal open fun checkIsMutable(): Unit {\n }\n\n /**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n /**\n * Based on GWT AbstractList\n * Copyright 2007 Google Inc.\n */\n\n package kotlin.collections\n\n /**\n * Provides a skeletal implementation of the [MutableList] interface.\n */\n @param E the type of elements contained in the list. The list is invariant in its element type.\n\n /**\n * public actual abstract class AbstractMutableList<E> protected actual constructor():\n AbstractMutableCollection<E>(), MutableList<E> {\n protected var modCount: Int = 0\n\n abstract override fun add(index: Int, element: E): Unit\n abstract override fun removeAt(index: Int): E\n abstract override fun set(index: Int, element: E): E\n\n /**\n * Adds the specified element to the end of this list.\n */\n @return `true` because the list is always modified as the result of this operation.\n\n actual override fun add(element: E): Boolean {\n checkIsMutable()\n add(size, element)\n return true\n }\n\n actual override fun addAll(index: Int, elements: Collection<E>): Boolean {\n AbstractList.checkPositionIndex(index, size)\n checkIsMutable()\n var _index = index\n var changed = false\n for (e in elements) {\n add(_index++, e)\n changed = true\n }\n return changed\n }\n\n actual override fun clear() {\n checkIsMutable()\n removeRange(0, size)\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return removeAll { it in elements }\n }\n\n actual override fun retainAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return removeAll { !it in elements }\n }\n\n actual override fun iterator(): MutableIterator<E> = IteratorImpl()\n\n actual override fun contains(element: E): Boolean = indexOf(element) >= 0\n\n actual override fun indexOf(element: E): Int {\n for (index in 0..lastIndex) {\n if (get(index) == element) {\n return index\n }\n }\n return -1\n }\n\n actual override fun lastIndexOf(element: E): Int {\n for (index in lastIndex downTo 0) {\n if (get(index) == element) {\n return index\n }\n }\n return -1\n }\n\n actual override fun listIterator(): MutableListIterator<E> = listIterator(0)\n\n actual override fun listIterator(index: Int): MutableListIterator<E> = ListIteratorImpl(index)\n\n actual override fun subList(fromIndex: Int, toIndex: Int): MutableList<E> = SubList(this, fromIndex, toIndex)\n\n /**\n * Removes the range of elements from this list starting from [fromIndex] and ending with but not including [toIndex].\n */\n protected open fun removeRange(fromIndex: Int, toIndex: Int) {\n val iterator = listIterator(fromIndex)\n repeat(toIndex - fromIndex) {\n iterator.next()\n iterator.remove()\n }\n }\n\n /**\n * Compares this list with another list instance with the ordered structural equality.\n */\n @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same order.\n\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is List<*>) return false\n return AbstractList.orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this list.\n */\n override fun hashCode(): Int = AbstractList.orderedHashCode(this)\n\n private open inner class IteratorImpl : MutableIterator<E> {\n /**\n * the index of the item that will be returned on the next call to [next]()\n */\n protected var index = 0\n }\n\n }

```

```

index of the item that was returned on the previous call to [next]() * or [ListIterator.previous]() (for
`ListIterator`),\n * -1 if no such item exists\n */\n protected var last = -1\n override fun
hasNext(): Boolean = index < size\n\n override fun next(): E {\n if (!hasNext()) throw NoSuchElementException()\n last = index++\n return get(last)\n }\n override fun remove() {\n check(last != -1) { \"Call next() or previous()
before removing element from the iterator.\" }\n removeAt(last)\n index = last\n last = -1\n }\n /**\n * Implementation of `MutableListIterator` for abstract lists.\n */\n private inner class
ListIteratorImpl(index: Int) : IteratorImpl(), MutableListIterator<E> {\n init {\n AbstractList.checkPositionIndex(index, this@AbstractMutableList.size)\n this.index = index\n }\n override fun hasPrevious(): Boolean = index > 0\n override fun nextIndex(): Int = index\n override fun
previous(): E {\n if (!hasPrevious()) throw NoSuchElementException()\n last = --index\n return get(last)\n }\n override fun previousIndex(): Int = index - 1\n override fun add(element: E) {\n add(index, element)\n index++\n last = -1\n }\n override fun set(element: E) {\n check(last != -1) { \"Call next() or previous() before updating element value with the iterator.\" }\n set(last,
element)\n }\n private class SubList<E>(private val list: AbstractMutableList<E>, private val
fromIndex: Int, toIndex: Int) : AbstractMutableList<E>(), RandomAccess {\n private var _size: Int = 0\n init {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, list.size)\n this._size = toIndex -
fromIndex\n }\n override fun add(index: Int, element: E) {\n AbstractList.checkPositionIndex(index, _size)\n list.add(fromIndex + index, element)\n _size++\n }\n override fun get(index: Int): E {\n AbstractList.checkElementIndex(index, _size)\n return list[fromIndex + index]\n }\n override fun removeAt(index: Int): E {\n AbstractList.checkElementIndex(index, _size)\n val result
= list.removeAt(fromIndex + index)\n _size--\n return result\n }\n override fun set(index:
Int, element: E): E {\n AbstractList.checkElementIndex(index, _size)\n return list.set(fromIndex +
index, element)\n }\n override val size: Int get() = _size\n internal override fun checkIsMutable():
Unit = list.checkIsMutable()\n }\n }\n /**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n /**\n * Based on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\n /**\n * Provides
a skeletal implementation of the [MutableMap] interface.\n */\n /**\n * The implementor is required to implement
[entries] property, which should return mutable set of map entries, and [put] function.\n */\n /**\n * @param K the type of
map keys. The map is invariant in its key type.\n */\n /**\n * @param V the type of map values. The map is invariant in its
value type.\n */\n public actual abstract class AbstractMutableMap<K, V> protected actual constructor() :
AbstractMap<K, V>(), MutableMap<K, V> {\n /**\n * A mutable [Map.Entry] shared by several [Map]
implementations.\n */\n internal open class SimpleEntry<K, V>(override val key: K, value: V) :
MutableMap.MutableEntry<K, V> {\n constructor(entry: Map.Entry<K, V>) : this(entry.key, entry.value)\n private var _value = value\n override val value: V get() = _value\n override fun setValue(newValue:
V): V {\n // Should check if the map containing this entry is mutable.\n // However, to not
increase entry memory footprint it might be worthwhile not to check it here and\n // force subclasses that
implement `build()` (freezing) operation to implement their own `MutableEntry`.\n this@AbstractMutableMap.checkIsMutable()\n val oldValue = this._value\n this._value = newValue\n return oldValue\n }\n override fun hashCode(): Int = entryHashCode(this)\n override fun
toString(): String = entryToString(this)\n override fun equals(other: Any?): Boolean = entryEquals(this,
other)\n }\n // intermediate abstract class to workaround KT-43321\n internal abstract class
AbstractEntrySet<E : Map.Entry<K, V>, K, V> : AbstractMutableSet<E>() {\n final override fun
contains(element: E): Boolean = containsEntry(element)\n abstract fun containsEntry(element: Map.Entry<K,
V>): Boolean\n final override fun remove(element: E): Boolean = removeEntry(element)\n abstract fun

```

```

removeEntry(element: Map.Entry<K, V>): Boolean\n }\n\n actual override fun clear() {\n entries.clear()\n }\n\n private var _keys: MutableSet<K>? = null\n actual override val keys: MutableSet<K>\n get() {\n if (_keys == null) {\n _keys = object : AbstractMutableSet<K>() {\n override fun\n add(element: K): Boolean = throw UnsupportedOperationException("Add is not supported on keys")\n }\n }\n override fun clear() {\n this@AbstractMutableMap.clear()\n }\n override\n operator fun contains(element: K): Boolean = containsKey(element)\n override operator fun iterator():\n MutableIterator<K> {\n val entryIterator = entries.iterator()\n return object :\n MutableIterator<K> {\n override fun hasNext(): Boolean = entryIterator.hasNext()\n\n override fun next(): K = entryIterator.next().key\n override fun remove() =\n entryIterator.remove()\n }\n }\n override fun remove(element: K): Boolean\n {\n checkIsMutable()\n if (containsKey(element)) {\n this@AbstractMutableMap.remove(element)\n return true\n }\n return\n false\n }\n override val size: Int get() = this@AbstractMutableMap.size\n override fun checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n }\n }\n return _keys!!\n }\n\n actual abstract override fun put(key: K, value: V): V?\n actual override fun\n putAll(from: Map<out K, V>) {\n checkIsMutable()\n for ((key, value) in from) {\n put(key,\n value)\n }\n }\n\n private var _values: MutableCollection<V>? = null\n actual override val values:\n MutableCollection<V>\n get() {\n if (_values == null) {\n _values = object :\n AbstractMutableCollection<V>() {\n override fun add(element: V): Boolean = throw\n UnsupportedOperationException("Add is not supported on values")\n override fun clear() =\n this@AbstractMutableMap.clear()\n override operator fun contains(element: V): Boolean =\n containsValue(element)\n override operator fun iterator(): MutableIterator<V> {\n val\n entryIterator = entries.iterator()\n return object : MutableIterator<V> {\n override fun\n hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): V = entryIterator.next().value\n override fun remove() = entryIterator.remove()\n }\n }\n override val size: Int get() = this@AbstractMutableMap.size\n }\n override fun checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n }\n }\n\n return _values!!\n }\n\n actual override fun remove(key: K): V? {\n checkIsMutable()\n val iter = entries.iterator()\n while (iter.hasNext()) {\n val entry = iter.next()\n val k = entry.key\n if (key == k) {\n val value = entry.value\n iter.remove()\n return value\n }\n }\n return null\n }\n\n /**\n * This method is called every time when a mutating method is called on\n * this mutable map.\n * Mutable maps that are built (frozen) must throw `UnsupportedOperationException`.\n */\n internal open fun checkIsMutable(): Unit {\n }\n\n /**\n * Copyright\n * 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed\n * by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n package\n kotlin.collections\n\n /**\n * Provides a skeletal implementation of the [MutableSet] interface.\n */\n @param E the\n type of elements contained in the set. The set is invariant in its element type.\n public actual abstract class\n AbstractMutableSet<E> protected actual constructor() : AbstractMutableCollection<E>(),\n MutableSet<E> {\n }\n\n /**\n * Compares this set with another set instance with the unordered structural equality.\n */\n @return\n `true`, if [other] instance is a [Set] of the same size, all elements of which are\n contained in this set.\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is Set<*>) return\n false\n return AbstractSet.setEquals(this, other)\n }\n\n /**\n * Returns\n * the hash code value for this set.\n */\n override fun hashCode(): Int =\n AbstractSet.unorderedHashCode(this)\n\n /**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming\n * Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the\n * license/LICENSE.txt file.\n */\n package kotlin.collections\n\n /**\n * Provides a [MutableList] implementation,\n
```

```

which uses a resizable array as its backing storage.\n * This implementation doesn't provide a way to manage
capacity, as backing JS array is resizeable itself.\n * There is no speed advantage to pre-allocating array sizes in
JavaScript, so this implementation does not include any of the\n * capacity and "growth increment" concepts.\n
*\npublic actual open class ArrayList<E> internal constructor(private var array: Array<Any?>) :
AbstractMutableList<E>(), MutableList<E>, RandomAccess {\n private var isReadOnly: Boolean = false\n\n
/**\n * Creates an empty [ArrayList].\n
*\n public actual constructor() : this(emptyArray()) {\n\n /**\n * Creates an empty [ArrayList].\n
*\n @param initialCapacity initial capacity (ignored)\n *\n public actual constructor(initialCapacity: Int) :
this(emptyArray()) {\n\n /**\n * Creates an [ArrayList] filled from the [elements] collection.\n
*\n public actual constructor(elements: Collection<E>) : this(elements.toTypedArray<Any?>()) {\n\n @PublishedApi\n
internal fun build(): List<E> {\n checkIsMutable()\n isReadOnly = true\n return this\n }\n\n /**
Does nothing in this ArrayList implementation.\n
*\n public actual fun trimToSize() {\n\n /** Does nothing in
this ArrayList implementation.\n
*\n public actual fun ensureCapacity(minCapacity: Int) {\n\n actual override
val size: Int get() = array.size\n @Suppress("UNCHECKED_CAST")\n actual override fun get(index: Int): E =
array[rangeCheck(index)] as E\n actual override
fun set(index: Int, element: E): E {\n checkIsMutable()\n rangeCheck(index)\n
@Suppress("UNCHECKED_CAST")\n return array[index].apply { array[index] = element } as E\n }\n\n
actual override fun add(element: E): Boolean {\n checkIsMutable()\n array.asDynamic().push(element)\n
modCount++\n return true\n }\n\n actual override fun add(index: Int, element: E): Unit {\n
checkIsMutable()\n array.asDynamic().splice(insertionRangeCheck(index), 0, element)\n modCount++\n
}\n\n actual override fun addAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n if
(elements.isEmpty()) return false\n array += elements.toTypedArray<Any?>()\n modCount++\n
return true\n }\n\n actual override fun addAll(index: Int, elements: Collection<E>): Boolean {\n
checkIsMutable()\n insertionRangeCheck(index)\n\n if (index == size) return addAll(elements)\n
if (elements.isEmpty()) return false\n when (index) {\n size -> return addAll(elements)\n 0 ->
array = elements.toTypedArray<Any?>() + array\n else -> array = array.copyOfRange(0,
index).asDynamic().concat(elements.toTypedArray<Any?>(), array.copyOfRange(index, size))\n }\n\n
modCount++\n return true\n }\n\n actual override fun removeAt(index: Int): E {\n checkIsMutable()\n
rangeCheck(index)\n modCount++\n return if (index == lastIndex)\n array.asDynamic().pop()\n
else\n array.asDynamic().splice(index, 1)[0]\n }\n\n actual override fun remove(element: E): Boolean {\n
checkIsMutable()\n for (index in array.indices) {\n if (array[index] == element) {\n
array.asDynamic().splice(index, 1)\n modCount++\n return true\n }\n }\n return
false\n }\n\n override
fun removeRange(fromIndex: Int, toIndex: Int) {\n checkIsMutable()\n modCount++\n
array.asDynamic().splice(fromIndex, toIndex - fromIndex)\n }\n\n actual override fun clear() {\n
checkIsMutable()\n array = emptyArray()\n modCount++\n }\n\n\n actual override fun
indexOf(element: E): Int = array.indexOf(element)\n\n actual override fun lastIndexOf(element: E): Int =
array.lastIndexOf(element)\n\n override fun toString() = arrayToString(array)\n\n
@Suppress("UNCHECKED_CAST")\n override fun <T> toArray(array: Array<T>): Array<T> {\n if
(array.size < size) {\n return toArray() as Array<T>\n }\n\n (this.array as
Array<T>).copyInto(array)\n\n if (array.size > size) {\n array[size] = null as T // null-terminate\n
}\n\n return array\n }\n\n override fun toArray(): Array<Any?> {\n return js("[\n]").slice.call(array)\n
}\n\n\n internal override
fun checkIsMutable() {\n if (isReadOnly) throw UnsupportedOperationException()\n }\n\n private fun
rangeCheck(index: Int) = index.apply {\n AbstractList.checkElementIndex(index, size)\n }\n\n private fun
insertionRangeCheck(index: Int) = index.apply {\n AbstractList.checkPositionIndex(index, size)\n }\n\n", "*/\n
*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage

```

```

kotlin.collections\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {\n if
(getStableSortingIsSupported()) {\n array.asDynamic().sort(comparison)\n } else {\n
mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))\n }\n}\n\ninternal fun
<T> sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {\n if (getStableSortingIsSupported())
{\n val comparison = { a: T, b: T -> comparator.compare(a, b) }\n array.asDynamic().sort(comparison)\n
} else {\n mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)\n }\n}\n\ninternal fun
<T> sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {\n if
(fromIndex < toIndex - 1) {\n mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex - 1, comparator)\n
 }\n}\n\ninternal fun <T : Comparable<T>> sortArray(array: Array<out T>) {\n if
(getStableSortingIsSupported()) {\n val comparison = { a: T, b: T -> a.compareTo(b) }\n
array.asDynamic().sort(comparison)\n } else {\n mergeSort(array.unsafeCast<Array<T>>(), 0,
array.lastIndex, naturalOrder())\n }\n}\n\nprivate var _stableSortingIsSupported: Boolean? = null\nprivate fun
getStableSortingIsSupported(): Boolean {\n _stableSortingIsSupported?.let { return it }\n
_stableSortingIsSupported = false\n\n val array = js("{}").unsafeCast<Array<Int>>()\n // known implementations may use stable sort for arrays of up to
512 elements\n // so we create slightly more elements to test stability\n for (index in 0 until 600)
array.asDynamic().push(index)\n val comparison = { a: Int, b: Int -> (a and 3) - (b and 3) }\n
array.asDynamic().sort(comparison)\n for (index in 1 until array.size) {\n val a = array[index - 1]\n val b
= array[index]\n if ((a and 3) == (b and 3) && a >= b) return false\n }\n _stableSortingIsSupported = true\n
return true\n}\n\nprivate fun <T> mergeSort(array: Array<T>, start: Int, endInclusive: Int, comparator:
Comparator<in T>) {\n val buffer = arrayOfNulls<Any?>(array.size).unsafeCast<Array<T>>()\n val result =
mergeSort(array, buffer, start, endInclusive, comparator)\n if (result !== array) {\n for (i in start..endInclusive)
array[i] = result[i]\n }\n}\n\n// Both start and end are inclusive indices.\nprivate
fun <T> mergeSort(array: Array<T>, buffer: Array<T>, start: Int, end: Int, comparator: Comparator<in T>):
Array<T> {\n if (start == end) {\n return array\n }\n val median = (start + end) / 2\n val left =
mergeSort(array, buffer, start, median, comparator)\n val right = mergeSort(array, buffer, median + 1, end,
comparator)\n val target = if (left === buffer) array else buffer\n // Merge.\n var leftIndex = start\n var
rightIndex = median + 1\n for (i in start..end) {\n when {\n leftIndex <= median && rightIndex <= end
-> {\n val leftValue = left[leftIndex]\n val rightValue = right[rightIndex]\n if
(comparator.compare(leftValue, rightValue) <= 0) {\n target[i] = leftValue\n leftIndex++\n
 } else {\n target[i] = rightValue\n rightIndex++\n }\n }\n }\n }\n leftIndex
<= median -> {\n target[i] = left[leftIndex]\n leftIndex++\n }\n else /* rightIndex
<= end */ -> {\n target[i] = right[rightIndex]\n rightIndex++\n Unit // TODO: Fix KT-
31506\n }\n}\n}\n}\n\nreturn target\n}", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\npackage
kotlin.collections\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@kotlin.js.JsName("
contentDeepHashCodeImpl")\ninternal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {\n if (this ==
null) return 0\n var result = 1\n for (element in this) {\n val elementHash = when {\n element == null
-> 0\n isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()\n
 element is UByteArray -> element.contentHashCode()\n element is UShortArray ->
element.contentHashCode()\n element is UIntArray -> element.contentHashCode()\n element is
ULongArray -> element.contentHashCode()\n else -> element.hashCode()\n }\n result = 31 * result + elementHash\n }\n return result\n}", /*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\ninternal interface
EqualityComparator {\n /**\n * Subclasses must override to return a value indicating\n * whether or not two

```



```

keys or values are equal.\n */\n abstract fun equals(value1: Any?, value2: Any?): Boolean\n\n /**\n * Subclasses must override to return the hash code of a given key.\n */\n abstract fun hashCode(value: Any?): Int\n\n object HashCode : EqualityComparator {\n override fun equals(value1: Any?, value2: Any?): Boolean = value1 == value2\n override fun hashCode(value: Any?): Int = value?.hashCode() ?: 0\n }\n}"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap] interface.\n * This implementation makes no guarantees regarding the order of enumeration of [keys], [values] and [entries] collections.\n * // Classes that extend HashMap and implement `build()` (freezing) operation\n * // have to make sure mutating methods check `checkIsMutable`.\n */\npublic actual\n\nopen class HashMap<K, V> : AbstractMutableMap<K, V>, MutableMap<K, V> {\n private inner class\n EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")\n override fun clear() {\n this@HashMap.clear()\n }\n override fun containsEntry(element: Map.Entry<K, V>): Boolean = this@HashMap.containsEntry(element)\n override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = internalMap.iterator()\n override fun removeEntry(element: Map.Entry<K, V>): Boolean {\n if (contains(element)) {\n this@HashMap.remove(element.key)\n }\n return true\n }\n override val size: Int get() = this@HashMap.size\n }\n\n /**\n * Internal implementation of the map: either string-based or hashcode-based.\n */\n private val internalMap: InternalMap<K, V>\n private val equality: EqualityComparator\n\n internal constructor(internalMap: InternalMap<K, V>) : super() {\n this.internalMap = internalMap\n this.equality = internalMap.equality\n }\n\n /**\n * Constructs an empty [HashMap] instance.\n */\n actual constructor(): this(InternalHashMap(EqualityComparator.HashCode))\n\n /**\n * Constructs an empty [HashMap] instance.\n * @param initialCapacity the initial capacity (ignored)\n * @param loadFactor the load factor (ignored)\n * @throws IllegalArgumentException if the initial capacity or load factor are negative\n */\n actual constructor(initialCapacity: Int, loadFactor: Float) : this() {\n // This implementation of HashMap has no need of load factors or capacities.\n require(initialCapacity >= 0) { "Negative initial capacity: $initialCapacity" }\n require(loadFactor >= 0) { "Non-positive load factor: $loadFactor" }\n }\n\n actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n /**\n * Constructs an instance of [HashMap] filled with the contents of the specified [original] map.\n */\n actual constructor(original: Map<out K, V>) : this() {\n this.putAll(original)\n }\n\n actual override fun clear() {\n internalMap.clear()\n }\n\n actual override fun structureChanged(this) {\n }\n\n actual override fun containsKey(key: K): Boolean = internalMap.containsKey(key)\n\n actual override fun containsValue(value: V): Boolean = internalMap.any { equality.equals(it.value, value) }\n\n private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? = null\n\n actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n get() {\n if (_entries == null) {\n _entries = createEntrySet()\n }\n return _entries!!\n }\n\n internal\n open fun createEntrySet():\n MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n actual override operator fun get(key: K): V? = internalMap.get(key)\n\n actual override fun put(key: K, value: V): V? = internalMap.put(key, value)\n\n actual override fun remove(key: K): V? = internalMap.remove(key)\n\n actual override val size: Int get() = internalMap.size\n}\n\n/**\n * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing them.\n */\npublic fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {\n return HashMap<String, V>(\n V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }\n)\n}"/*\n * Copyright 2010-2018

```





```

contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n * Based on GWT InternalStringMap\n * Copyright 2008 Google Inc.\n
*\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * A simple wrapper
around JavaScript Map for key type is string.\n *\n * Though this map is instantiated only with K=String, the K type
is not fixed to String statically,\n * because we want to have it erased to Any? in order not to generate type-safe
override bridges for\n * [get], [contains], [remove] etc, if they ever are generated.\n
*\ninternal class InternalStringMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n\n
private var backingMap: dynamic = createJsMap()\n override var size: Int = 0\n private set\n\n// /**\n// *
A mod count to track 'value' replacements in map to ensure that the 'value' that we have in the\n// * iterator entry
is guaranteed to be still correct.\n// * This is to optimize for the common scenario where the values are not
modified during\n// * iterations where the entries are never stale.\n// */\n private var valueMod: Int = 0\n\n
override operator fun contains(key: K): Boolean {\n if (key !is String) return false\n return
backingMap[key] !== undefined\n }\n\n override operator fun get(key: K): V? {\n if (key !is String) return
null\n val value = backingMap[key]\n return if (value !== undefined) value.unsafeCast<V>() else null\n
}\n\n\n override fun put(key: K, value: V): V?
{\n require(key is String)\n val oldValue = backingMap[key]\n backingMap[key] = value\n\n if
(oldValue === undefined) {\n size++\n// structureChanged(host)\n return null\n } else {\n//
valueMod++\n return oldValue.unsafeCast<V>()\n }\n }\n\n override fun remove(key: K): V?
{\n if (key !is String) return null\n val value = backingMap[key]\n if (value !== undefined) {\n
jsDeleteProperty(backingMap, key)\n size--\n// structureChanged(host)\n return
value.unsafeCast<V>()\n } else {\n// valueMod++\n return null\n }\n }\n\n\n override fun
clear() {\n backingMap = createJsMap()\n size = 0\n }\n\n\n override fun iterator():
MutableIterator<MutableEntry<K, V>> {\n return object : MutableIterator<MutableEntry<K, V>> {\n
private val keys: Array<String>
= js("Object").keys(backingMap)\n private val iterator = keys.iterator()\n private var lastKey:
String? = null\n\n override fun hasNext(): Boolean = iterator.hasNext()\n\n override fun next():
MutableEntry<K, V> {\n val key = iterator.next()\n lastKey = key\n
@Suppress("UNCHECKED_CAST")\n return newMapEntry(key as K)\n }\n\n\n override
fun remove() {\n @Suppress("UNCHECKED_CAST")\n
this@InternalStringMap.remove(checkNotNull(lastKey) as K)\n }\n }\n }\n\n private fun
newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {\n override val key: K get() =
key\n override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()\n\n override fun
setValue(newValue: V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()\n\n override fun
hashCode(): Int
= AbstractMap.entryHashCode(this)\n\n override fun toString(): String = AbstractMap.entryToString(this)\n
override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this, other)\n }\n}\n", /*\n * Copyright
2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n * Based on GWT
LinkedHashMap\n * Copyright 2008 Google Inc.\n */\npackage kotlin.collections\n\nimport
kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap]
interface, which additionally preserves the insertion order\n * of entries during the iteration.\n *\n * The insertion
order is preserved by maintaining a doubly-linked list of all of its entries.\n */\n\npublic actual open class
LinkedHashMap<K, V> : HashMap<K, V>, MutableMap<K, V> {\n\n // **\n * The entry we use includes
next/prev pointers for a doubly-linked
circular\n * list with a head node. This reduces the special cases we have to deal with\n * in the list
operations.\n\n * Note that we duplicate the key from the underlying hash map so we can find\n * the eldest
entry. The alternative would have been to modify HashMap so more\n * of the code was directly usable here, but
this would have added some\n * overhead to HashMap, or to reimplement most of the HashMap code here with\n

```

```

* small modifications. Paying a small storage cost only if you use * LinkedHashMap and minimizing code size
seemed like a better tradeoff
*/
private inner class ChainEntry<K, V>(key: K, value: V) :
AbstractMutableMap.SimpleEntry<K, V>(key, value) {
 internal var next: ChainEntry<K, V>? = null
 internal var prev: ChainEntry<K, V>? = null
 override fun setValue(newValue: V): V {
 this@LinkedHashMap.checkIsMutable()
 return super.setValue(newValue)
 }
}
private inner class EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V> {
 private inner
class EntryIterator : MutableIterator<MutableEntry<K, V>> {
 // The last entry that was returned from this
iterator.
 private var last: ChainEntry<K, V>? = null
 // The next entry to return from this
iterator.
 private var next: ChainEntry<K, V>? = null
 init {
 next = head
 }
 recordLastKnownStructure(map, this)
 override fun hasNext(): Boolean {
 return
next != null
 }
 override fun next(): MutableEntry<K, V> {
 checkStructuralChange(map, this)
 if (!hasNext()) throw NoSuchElementException()
 val
current = next!!
 last = current
 next = current.next.takeIf { it != head }
 return
current
 }
 override fun remove() {
 check(last != null)
 this@EntrySet.checkIsMutable()
 checkStructuralChange(map, this)
 last!!.remove()
 map.remove(last!!.key)
 recordLastKnownStructure(map, this)
 last = null
 }
 override fun add(element:
MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")
 override fun clear() {
 this@LinkedHashMap.clear()
 }
 override fun containsEntry(element:
Map.Entry<K, V>): Boolean = this@LinkedHashMap.containsEntry(element)
 override operator fun
iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()
 override fun removeEntry(element:
Map.Entry<K, V>): Boolean {
 checkIsMutable()
 if (contains(element)) {
 this@LinkedHashMap.remove(element.key)
 }
 return true
 }
 return false
 }
 override val size: Int get() =
this@LinkedHashMap.size
 override fun checkIsMutable(): Unit =
this@LinkedHashMap.checkIsMutable()
}
/*
 * The head of the insert order chain, which is a doubly-
linked circular
 * list.
 * The most recently inserted node is at the end of the chain, ie.
 * chain.prev.
*/
private var head: ChainEntry<K, V>? = null
/**
 * Add this node to the end of the chain.
*/
private fun ChainEntry<K, V>.addToEnd() {
 // This entry is not in the list.
 check(next == null && prev
== null)
 val _head = head
 if (_head == null) {
 head = this
 next = this
 prev =
this
 } else {
 // Chain is valid.
 val _tail = checkNotNull(_head.prev)
 // Update me.
 prev = _tail
 next = _head
 // Update my new siblings: current head and old tail
 _head.prev = this
 _tail.next = this
 }
}
/**
 * Remove this node from the chain it is a part of.
*/
private fun ChainEntry<K,
V>.remove() {
 if (this.next === this) {
 // if this is single element, remove head
 head = null
 } else {
 if (head === this) {
 // if this is first element, move head to next
 head =
next
 }
 next!!.prev = prev
 prev!!.next = next
 next = null
 prev = null
 }
}
/*
 * The hashmap that keeps track of our entries and the chain. Note that we
 * duplicate the key here
to eliminate changes to HashMap and minimize the
 * code here, at the expense of additional space.
*/
private val map: HashMap<K, ChainEntry<K, V>>
private var isReadOnly: Boolean = false
/**
 * Constructs an empty [LinkedHashMap] instance.
*/
actual constructor(): super() {
 map =
HashMap<K, ChainEntry<K, V>>()
}
internal constructor(backingMap: HashMap<K, Any>): super() {
 @Suppress("UNCHECKED_CAST") // expected to work due to erasure
 map = backingMap as
HashMap<K, ChainEntry<K, V>>
}
/**
 * Constructs an empty [LinkedHashMap] instance.
 *
 * @param initialCapacity the initial capacity (ignored)
 * @param loadFactor the load factor (ignored)
 *
 * @throws IllegalArgumentException if the initial capacity or load factor are negative
*/
actual
constructor(initialCapacity: Int, loadFactor: Float): super(initialCapacity, loadFactor) {
 map = HashMap<K,
ChainEntry<K, V>>()
}
actual constructor(initialCapacity: Int): this(initialCapacity, 0.0f)
/**
 *

```



InvocationKind.EXACTLY\_ONCE)\n

```
\n return block()\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.io\n\ninternal abstract class BaseOutput {\n open fun println()\n print("\\n")\n}\n\nopen fun println(message: Any?) {\n print(message)\n println()\n}\n\nabstract fun print(message: Any?)\n\nopen fun flush() {\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n\n@JsName("NodeJsOutput")\ninternal class NodeJsOutput(val outputStream: dynamic) : BaseOutput() {\n override fun print(message: Any?) {\n // TODO: Using local variable because of bug in block decomposition lowering in IR backend\n val messageString = String(message)\n outputStream.write(messageString)\n }\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n\n@JsName("OutputToConsoleLog")\ninternal class OutputToConsoleLog : BaseOutput() {\n override fun print(message: Any?) {\n console.log(message)\n }\n\n override fun println(message: Any?) {\n console.log(message)\n }\n\n override fun println() {\n console.log("")\n }\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */\n\n@JsName("BufferedOutput")\ninternal open class BufferedOutput : BaseOutput() {\n var buffer = ""\n\n override fun print(message: Any?) {\n buffer += String(message)\n }\n\n override fun flush() {\n buffer = ""\n }\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n\n@JsName("BufferedOutputToConsoleLog")\ninternal class BufferedOutputToConsoleLog : BufferedOutput() {\n override fun print(message: Any?) {\n var s = String(message)\n val i = s.nativeLastIndexOf("\\n", 0)\n if (i >= 0) {\n buffer += s.substring(0, i)\n flush()\n s = s.substring(i + 1)\n }\n buffer += s\n }\n\n override fun flush() {\n console.log(buffer)\n buffer = ""\n }\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */\n\n@JsName("output")\ninternal var output = run {\n val isNode: Boolean = js("typeof process !== 'undefined' && process.versions && !process.versions.node")\n if (isNode) NodeJsOutput(js("process.stdout"))\n else BufferedOutputToConsoleLog()\n}\n\n\n@kotlin.internal.InlineOnly\nprivate inline fun String(value: Any?): String = js("String")(value)\n\n/** Prints the line separator to the standard output stream. */\n\npublic actual fun println() {\n output.println()\n}\n\n/** Prints the given [message] and the line separator to the standard output stream. */\n\npublic actual fun println(message: Any?) {\n output.println(message)\n}\n\n/** Prints the given [message] to the standard output stream. */\n\npublic actual fun print(message: Any?) {\n output.print(message)\n}\n\n\n@SinceKotlin("1.6")\npublic actual fun readln(): String = throw UnsupportedOperationException("readln is not supported in Kotlin/JS")\n\n\n@SinceKotlin("1.6")\npublic actual fun readlnOrNull(): String? = throw UnsupportedOperationException("readlnOrNull is not supported in Kotlin/JS")\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal actual class SafeContinuation<in T>\n\ninternal actual constructor(\n private val delegate: Continuation<T>,\n initialResult: Any?) : Continuation<T> {\n @PublishedApi\n internal actual constructor(delegate: Continuation<T>) : this(delegate, UNDECIDED)\n\n public actual override val context: CoroutineContext\n get() = delegate.context\n\n private var result: Any? = initialResult\n\n public actual override fun resumeWith(result: Result<T>) {\n val cur = this.result\n when {\n cur === UNDECIDED -> {\n this.result = result.value\n }\n cur === COROUTINE_SUSPENDED -> {\n this.result = RESUMED\n delegate.resumeWith(result)\n }\n else -> throw IllegalStateException("Already resumed")\n }\n }\n\n @PublishedApi\n internal actual fun getOrThrow(): Any? {\n if (result === UNDECIDED) {\n result = COROUTINE_SUSPENDED\n return COROUTINE_SUSPENDED\n }\n val result = this.result\n return when {\n
```

```

 result === RESUMED -> COROUTINE_SUSPENDED // already called continuation, indicate
COROUTINE_SUSPENDED upstream\n result is Result.Failure -> throw result.exception\n else ->
result // either COROUTINE_SUSPENDED or data\n }\n }\n}\n", /*\n * Copyright 2010-2020 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.coroutines.cancellation\n\n@SinceKotlin("1.4")\npublic actual open class CancellationException :
IllegalStateException {\n actual constructor() : super()\n actual constructor(message: String?) : super(message)\n
 constructor(message: String?, cause: Throwable?) : super(message, cause)\n constructor(cause: Throwable?) :
super(cause)\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines.js.internal\n\nimport
kotlin.coroutines.Continuation\nimport
kotlin.coroutines.EmptyCoroutineContext\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal val
EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result ->\n result.getOrThrow()\n}\n", /*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n/**\n * Exposes the [Date API](https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.\n
*/\n\n@Suppress("NOT_DOCUMENTED")\npublic external class Date() {\n public constructor(milliseconds:
Number)\n\n public constructor(dateString: String)\n\n public constructor(year: Int, month: Int)\n\n public
constructor(year: Int, month: Int, day: Int)\n\n public
constructor(year: Int, month: Int, day: Int, hour: Int)\n\n public constructor(year: Int, month: Int, day: Int, hour:
Int, minute: Int)\n\n public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int)\n\n
public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number)\n\n
public fun getDate(): Int\n\n public fun getDay(): Int\n\n public fun getFullYear(): Int\n\n public fun
getHours(): Int\n\n public fun getMilliseconds(): Int\n\n public fun getMinutes(): Int\n\n public fun
getMonth(): Int\n\n public fun getSeconds(): Int\n\n public fun getTime(): Double\n\n public fun
getTimezoneOffset(): Int\n\n public fun getUTCDate(): Int\n\n public fun getUTCDay(): Int\n\n public fun
getUTCFullYear(): Int\n\n public fun getUTCHours(): Int\n\n public fun getUTCMilliseconds(): Int\n\n public
fun getUTCMinutes(): Int\n\n public fun getUTCMonth(): Int\n\n public fun
getUTCSeconds(): Int\n\n public fun toDateString(): String\n\n public fun toISOString(): String\n\n public fun
toJSON(): Json\n\n public fun toLocaleDateString(locales: Array<String> = definedExternally, options:
LocaleOptions = definedExternally): String\n\n public fun toLocaleDateString(locales: String, options:
LocaleOptions = definedExternally): String\n\n public fun toLocaleString(locales: Array<String> =
definedExternally, options: LocaleOptions = definedExternally): String\n\n public fun toLocaleString(locales:
String, options: LocaleOptions = definedExternally): String\n\n public fun toLocaleTimeString(locales:
Array<String> = definedExternally, options: LocaleOptions = definedExternally): String\n\n public fun
toLocaleTimeString(locales: String, options: LocaleOptions = definedExternally): String\n\n public fun
toTimeString(): String\n\n public fun toUTCString(): String\n\n public companion object {\n public fun
now(): Double\n\n public fun parse(dateString: String): Double\n\n public fun UTC(year: Int, month: Int): Double\n\n
public fun UTC(year: Int, month: Int, day: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour:
Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int): Double\n\n public
fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int): Double\n\n public fun UTC(year:
Int, month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number): Double\n }\n\n public
interface LocaleOptions {\n public var localeMatcher: String?\n\n public var timeZone: String?\n\n
public var hour12: Boolean?\n\n public var formatMatcher: String?\n\n public var weekday: String?\n\n
public var era: String?\n\n public var year: String?\n\n public var month: String?\n\n public var day:

```



```

String?
 public var hour: String?
 public var minute: String?
 public var second: String?
 public var timeZoneName: String?
 }
 public inline fun dateLocaleOptions(init: Date.LocaleOptions.() -> Unit): Date.LocaleOptions {
 val result = js("new Object()").unsafeCast<Date.LocaleOptions>()
 init(result)
 return result
 }
}
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.dom
import org.w3c.dom.Document
import org.w3c.dom.Element
import kotlin.internal.LowPriorityInOverloadResolution
import kotlinx.dom.appendElement as newAppendElement
import kotlinx.dom.createElement as newCreateElement
/*
 * Creates a new element with the specified [name].
 * The element is initialized with the specified [init] function.
 */
@LowPriorityInOverloadResolution
@Deprecated(
 message = "This API is moved to another package, use 'kotlinx.dom.createElement' instead.",
 replaceWith = ReplaceWith("this.createElement(name, init)"),
 kotlinSince = "1.4",
 errorSince = "1.6")
public inline fun Document.createElement(name: String, noinline init: Element.() -> Unit): Element =
 this.newCreateElement(name, init)
/*
 * Appends a newly created element with the specified [name] to this element.
 * The element is initialized with the specified [init] function.
 */
@LowPriorityInOverloadResolution
@Deprecated(
 message = "This API is moved to another package, use 'kotlinx.dom.appendElement' instead.",
 replaceWith = ReplaceWith("this.appendElement(name, init)"),
 kotlinSince = "1.4",
 errorSince = "1.6")
public inline fun Element.appendElement(name: String, noinline init: Element.() -> Unit): Element =
 this.newAppendElement(name, init)
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.dom
import org.w3c.dom.Element
import kotlin.internal.LowPriorityInOverloadResolution
import kotlinx.dom.addClass as newAddClass
import kotlinx.dom.hasClass as newHasClass
import kotlinx.dom.removeClass as newRemoveClass
/*
 * Returns true if the element has the given CSS class style in its 'class' attribute
 */
@LowPriorityInOverloadResolution
@Deprecated(
 message = "This API is moved to another package, use 'kotlinx.dom.hasClass' instead.",
 replaceWith = ReplaceWith("this.hasClass(cssClass)"),
 kotlinSince = "1.4",
 errorSince = "1.6")
inline fun Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)
/*
 * Adds CSS class to element. Has no effect if all specified classes are already in class attribute of the element
 */
@LowPriorityInOverloadResolution
@Deprecated(
 message = "This API is moved to another package, use 'kotlinx.dom.addClass' instead.",
 replaceWith = ReplaceWith("this.addClass(cssClasses)"),
 kotlinSince = "1.4",
 errorSince = "1.6")
inline fun Element.addClass(vararg cssClasses: String): Boolean = this.newAddClass(*cssClasses)
/*
 * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element
 */
@LowPriorityInOverloadResolution
@Deprecated(
 message = "This API is moved to another package, use 'kotlinx.dom.removeClass' instead.",
 replaceWith = ReplaceWith("this.removeClass(cssClasses)"),
 kotlinSince = "1.4",
 errorSince = "1.6")
inline fun Element.removeClass(vararg cssClasses: String): Boolean =
 this.newRemoveClass(*cssClasses)
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.dom
import org.w3c.dom.Element
import org.w3c.dom.Node
import kotlin.internal.LowPriorityInOverloadResolution
import kotlinx.dom.isElement as

```

```

newIsElement\nimport kotlin.dom.isText as newIsText\n\n/**\n * Gets a value indicating whether this node is a
TEXT_NODE or a CDATA_SECTION_NODE.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n
message = \"This API is moved to another package, use 'kotlin.dom.isText' instead.\",\n replaceWith =
ReplaceWith(\"this.isText\", \"kotlin.dom.isText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.6\")\npublic val Node.isText: Boolean\n
 inline get() = this.newIsText\n\n/**\n * Gets a value indicating whether this node is an [Element].\n
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package,
use 'kotlin.dom.isElement' instead.\",\n replaceWith = ReplaceWith(\"this.isElement\",
\"kotlin.dom.isElement\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic val
Node.isElement: Boolean\n inline get() = this.newIsElement\n\n\", \"/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom.events\n\npublic fun
EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)\n\nprivate class
EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {\n public override fun
handleEvent(event: Event) {\n handler(event)\n
 }\n\n public override fun toString(): String = \"EventListenerHandler($handler)\"\n}\n\n\", \"/*\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom\n\npublic
external interface ItemArrayLike<out T> {\n val length: Int\n fun item(index: Int): T?\n}\n\n/**\n * Returns the
view of this `ItemArrayLike<T>` collection as `List<T>`\n */\n@public fun <T> ItemArrayLike<T>.asList(): List<T>
= object : AbstractList<T>() {\n override val size: Int get() = this@asList.length\n\n override fun get(index: Int):
T = when (index) {\n in 0..lastIndex -> this@asList.item(index).unsafeCast<T>()\n else -> throw
IndexOutOfBoundsException(\"index $index is not in range [0..$lastIndex]\")\n }\n}\n\n\", \"/*\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source
code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlin.dom.appendText as newAppendText\nimport
kotlin.dom.clear as newClear\n\n/**\n * Removes all the children from this node.\n
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package,
use 'kotlin.dom.clear' instead.\",\n replaceWith = ReplaceWith(\"this.clear()\",
\"kotlin.dom.clear\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic inline fun
Node.clear() = this.newClear()\n\n/**\n * Creates text node and append it to the element.\n */\n@return this
element\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another
package, use 'kotlin.dom.appendText' instead.\",\n replaceWith = ReplaceWith(\"this.appendText(text)\",
\"kotlin.dom.appendText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\n\ninline fun
Element.appendText(text: String): Element = this.newAppendText(text)\n\n\", \"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Reinterprets this value
as a value of the [dynamic type](/docs/reference/dynamic-type.html).\n */\n@kotlin.internal.InlineOnly\npublic
inline fun Any?.asDynamic(): dynamic = this\n\n/**\n * Reinterprets this value as a value of the specified type [T]
without any actual type checking.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Any?.unsafeCast():
@kotlin.internal.NoInfer T = this.asDynamic()\n\n/**\n * Reinterprets this `dynamic` value as a value of the
specified type [T] without any actual type checking.\n
*/\n@kotlin.internal.DynamicExtension\n@JsName(\"unsafeCastDynamic\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this\n\n/**\n * Allows to iterate this `dynamic`
object in the following cases:\n * - when it has an `iterator` function,\n * - when it is an array\n * - when it is an
instance of [kotlin.collections.Iterable]\n */\n@kotlin.internal.DynamicExtension\npublic operator fun
dynamic.iterator(): Iterator<dynamic> {\n val r: Any? = this\n\n return when {\n this[\"iterator\"] != null -

```

```

>\n this["iterator"]()\n isArrayish(r) ->\n r.unsafeCast<Array<*>>().iterator()\n else ->\n (r as Iterable<*>).iterator()\n }\n }", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.enums\n\n// Unused stub\ninternal actual class
EnumEntriesSerializationProxy<E
: Enum<E>> actual constructor(entries: Array<E>)\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the
module\n\n@JsName("throwNPE")\ninternal fun throwNPE(message: String) {\n throw
NullPointerException(message)\n}\n\n@JsName("throwCCE")\ninternal fun throwCCE() {\n throw
ClassCastException("Illegal cast")\n}\n\n@JsName("throwISE")\ninternal fun throwISE(message: String) {\n
throw IllegalStateException(message)\n}\n\n@JsName("throwUPAE")\ninternal fun throwUPAE(propertyName:
String) {\n throw UninitializedPropertyAccessException("lateinit property ${propertyName} has not been
initialized")\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.collections\n\n/**\n * Groups elements from the [Grouping] source by key and counts elements in each
group.\n * @return a [Map] associating the key of each group with the count of elements in the group.\n * @sample
samples.collections.Grouping.groupingByEachCount\n */\n\n@SinceKotlin("1.1")\npublic actual fun <T,
K> Grouping<T, K>.eachCount(): Map<K, Int> =\n fold(0) { acc, _ -> acc + 1 }\n\n/**\n * Groups elements
from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements in each
group.\n * @return a [Map] associating the key of each group with the count of element in the group.\n */\n\n@SinceKotlin("1.1")\npublic inline fun <T, K> Grouping<T, K>.eachSumOf(valueSelector: (T) -> Int):
Map<K, Int> =\n fold(0) { acc, e -> acc + valueSelector(e) }\n", /*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("GroupingKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
kotlin.collections\n\n/**\n * Represents a source of elements with a [keyOf] function, which can be applied to each
element to get its key.\n * A [Grouping] structure serves as an intermediate step in group-and-fold operations:\n
* they group elements by their keys and then fold each group with some aggregating operation.\n * It is created
by attaching `keySelector: (T) -> K` function to a source of elements.\n * To get an instance of [Grouping] use one
of `groupingBy` extension functions:\n * - [Iterable.groupingBy]\n * - [Sequence.groupingBy]\n * -
[Array.groupingBy]\n * - [CharSequence.groupingBy]\n */\n * For the list of group-and-fold operations available,
see the [extension functions](#extension-functions) for `Grouping`.\n */\n\n@SinceKotlin("1.1")\npublic
interface Grouping<T, out K> {\n /** Returns an [Iterator] over the elements of the source of this grouping. */\n
fun sourceIterator(): Iterator<T>\n /** Extracts the key of an [element]. */\n fun keyOf(element: T):
K}\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each
group sequentially,\n * passing the previously accumulated value and the current element as arguments, and stores
the results in a new map.\n * The key for each element is provided by the [Grouping.keyOf] function.\n * @param
operation function is invoked on each element with the following parameters:\n * - `key`: the key of the
group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group, can be `null`
if it's the first `element` encountered in the group;\n * - `element`: the element from the source being aggregated;\n
* - `first`: indicates whether it's the first `element` encountered
in the group.\n * @return a [Map] associating the key of each group with the result of aggregation of the group
elements.\n * @sample samples.collections.Grouping.aggregateByRadix\n */\n\n@SinceKotlin("1.1")\npublic inline
fun <T, K, R> Grouping<T, K>.aggregate(\n operation: (key: K, accumulator: R?, element: T, first: Boolean) ->
R\n): Map<K, R> {\n return aggregateTo(mutableMapOf<K, R>(), operation)\n}\n\n/**\n * Groups elements

```

from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map. The key for each element is provided by the [Grouping.keyOf] function. @param operation a function that is invoked on each element with the following parameters: `key`: the key of the group this element belongs to; `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group; `element`: the element from the source being aggregated; `first`: indicates whether it's the first `element` encountered in the group. If the [destination] map already has a value corresponding to some key, then the elements being aggregated for that key are never considered as `first`. @return the [destination] map associating the key of each group with the result of aggregation of the group elements. @sample samples.collections.Grouping.aggregateByRadixTo

```

*\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.aggregateTo(\n destination: M,\n operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R)\n): M {\n for (e in this.sourceIterator()) {\n val key = keyOf(e)\n val accumulator = destination[key]\n destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))\n }\n return destination\n}\n
```

\* Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. An initial value of accumulator is provided by [initialValueSelector] function. @param initialValueSelector a function that provides an initial value of accumulator for each group. It's invoked with parameters: `key`: the key of the group; `element`: the first element being encountered in that group. @param operation a function that is invoked on each element with the following parameters: `key`: the key of the group this element belongs to; `accumulator`: the current value of the accumulator of the group; `element`: the element from the source being accumulated. @return a [Map] associating the key of each group with the result of accumulating the group elements. @sample

```

samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue\n *\n@SinceKotlin("1.1")\npublic inline fun <T, K, R> Grouping<T, K>.fold(\n initialValueSelector: (key: K, element: T) -> R,\n operation: (key: K, accumulator: R, element: T) -> R)\n): Map<K, R> =\n @Suppress("UNCHECKED_CAST")\n aggregate {\n key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }\n
```

\* Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map. An initial value of accumulator is provided by [initialValueSelector] function. @param initialValueSelector a function that provides an initial value of accumulator for each group. It's invoked with parameters: `key`: the key of the group; `element`: the first element being encountered in that group. If the [destination] map already has a value corresponding to some key, that value is used as an initial value of the accumulator for that group and the [initialValueSelector] function is not called for that group. @param operation a function that is invoked on each element with the following parameters: `key`: the key of the group this element belongs to; `accumulator`: the current value of the accumulator of the group; `element`: the element from the source being accumulated. @return the [destination] map associating the key of each group with the result of accumulating the group elements. @sample

```

samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo\n *\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n destination: M,\n initialValueSelector: (key: K, element: T) -> R,\n operation: (key: K, accumulator: R, element: T) -> R)\n): M =\n @Suppress("UNCHECKED_CAST")\n aggregateTo(destination) {\n key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }\n
```

\* Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. An initial value of accumulator is the same [initialValue] for each group.

@param operation a function that is invoked on each element with the following parameters:  
`accumulator`: the current value of the accumulator of the group;  
`element`: the element from the source being accumulated.  
@return a [Map] associating the key of each group with the result of accumulating the group elements.  
@sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue

```

*\n@SinceKotlin("1.1")\npublic
inline fun <T, K, R> Grouping<T, K>.fold(\n initialValue: R,\n operation: (accumulator: R, element: T) ->
R\n): Map<K, R> =\n @Suppress("UNCHECKED_CAST")\n aggregate { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies
[operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current
element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is
the same [initialValue] for each group.\n *\n * If the [destination] map already has a value corresponding to the key
of some group,\n * that value is used as an initial value of the accumulator for that group.\n *\n * @param operation
a function that is invoked on each element with the following parameters:

`accumulator`: the current value of the accumulator of the group;

`element`: the element from the source being accumulated.

@return the [destination] map associating the key of each group with the
result of accumulating the group elements.\n * @sample
samples.collections.Grouping.foldByEvenLengthWithConstantInitialValueTo\n *\n@SinceKotlin("1.1")\npublic
inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n destination: M,\n initialValue: R,\n operation: (accumulator: R, element: T) -> R\n): M =\n @Suppress("UNCHECKED_CAST")\n aggregateTo(destination) { _, acc, e, first -> operation(if (first) initialValue else acc as R, e) }\n\n\n/**\n * Groups
elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group\n *
sequentially starting from the second element of the group,\n * passing the previously accumulated value and the
current element as arguments,\n * and stores the results in a new map.\n * An initial value of accumulator is the first
element of the group.\n *\n * @param operation a
function that is invoked on each subsequent element of the group with the following parameters:

`key`: the key of the group this element belongs to;

`accumulator`: the current value of the accumulator of the group;

`element`: the element from the source being accumulated.

@return a [Map] associating the key of each group with the result of accumulating the group elements.\n * @sample
samples.collections.Grouping.reduceByMaxVowels\n *\n@SinceKotlin("1.1")\npublic inline fun <S, T : S, K>
Grouping<T, K>.reduce(\n operation: (key: K, accumulator: S, element: T) -> S\n): Map<K, S> =\n aggregate {
key, acc, e, first ->\n @Suppress("UNCHECKED_CAST")\n if (first) e else operation(key, acc as S, e)\n }\n\n\n/**\n * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the
elements of each group\n * sequentially starting from the second element of the group,\n * passing the previously
accumulated value and the current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is the first element of the group.\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an initial value of the accumulator for that group and the first
element of that group is also\n * subjected to the [operation].\n *\n * @param operation a function that is invoked on
each subsequent element of the group with the following parameters:

`accumulator`: the current value of the accumulator of the group;

`element`: the element from the source being folded;\n *\n * @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n *
@sample samples.collections.Grouping.reduceByMaxVowelsTo\n *\n@SinceKotlin("1.1")\npublic inline fun <S,
T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(\n destination: M,\n operation:
(key: K, accumulator: S, element: T) -> S\n): M =\n aggregateTo(destination) { key, acc, e, first ->\n @Suppress("UNCHECKED_CAST")\n if (first) e else operation(key, acc as S, e)\n }\n\n\n/**\n * Groups
elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an
initial value of the counter for that group.\n *\n * @return the [destination] map associating the key of each group

```



```

fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: String): String\n
public fun stringify(o: Any?, replacer: Array<String>): String\n public fun stringify(o: Any?, replacer:
Array<String>, space: Int): String\n public fun stringify(o: Any?, replacer: Array<String>, space: String):
String\n\n public fun <T> parse(text: String): T\n public fun <T> parse(text: String, reviver: ((key: String, value:
Any?) -> Any?)): T\n}\n"/\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use

```

of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```

*\npackage kotlin.math\n\nimport kotlin.internal.InlineOnly\nimport kotlin.js.JsMath as nativeMath\n\n//
region ===== Double Math =====\n\n**

```

Computes the sine of the angle [x] given in radians.\n \* Special cases:\n \* - `sin(NaN|+Inf|-Inf)` is `NaN`\n

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sin(x: Double): Double =

```

nativeMath.sin(x)\n\n\*\* Computes the cosine of the angle [x] given in radians.\n \* Special cases:\n \* -  
`cos(NaN|+Inf|-Inf)` is `NaN`\n \* \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cos(x: Double):

```

Double = nativeMath.cos(x)\n\n** Computes the tangent of the angle [x] given in radians.\n * Special cases:\n
* - `tan(NaN|+Inf|-Inf)` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tan(x:

```

```

Double): Double = nativeMath.tan(x)\n\n**\n

```

\* Computes the arc sine of the value [x];\n \* the returned value is an angle in the range from `-PI/2` to `PI/2`  
radians.\n \* Special cases:\n \* - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun asin(x: Double): Double =

```

nativeMath.asin(x)\n\n\*\*\n \* Computes the arc cosine of the value [x];\n \* the returned value is an angle in the  
range from `0.0` to `PI` radians.\n \* Special cases:\n \* - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun acos(x: Double): Double =

```

nativeMath.acos(x)\n\n\*\*\n \* Computes the arc tangent of the value [x];\n \* the returned value is an angle in the  
range from `-PI/2` to `PI/2` radians.\n \* Special cases:\n \* - `atan(NaN)` is `NaN`\n

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atan(x: Double): Double =

```

```

nativeMath.atan(x)\n\n**\n * Returns the angle `theta` of the polar

```

coordinates `(r, theta)` that correspond\n \* to the rectangular coordinates `(x, y)` by computing the arc tangent of  
the value [y] / [x];\n \* the returned value is an angle in the range from `-PI` to `PI` radians.\n \* Special cases:\n \*

```

- `atan2(0.0, 0.0)` is `0.0`\n * - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`\n * - `atan2(-0.0, x)` is `
0.0` for `x > 0` and `-PI` for `x < 0`\n * - `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for `-Inf < y < 0`\n *

```

```

- `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n * - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `
-PI/2` for `y < 0`\n * - `atan2(+Inf, x)` is `PI/2` for finite `x`y\n * - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n * -

```

```

`atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
atan2(y: Double, x: Double): Double = nativeMath.atan2(y, x)\n\n**\n * Computes the hyperbolic sine of the value

```

```

[x].\n * Special

```

```

cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n

```

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sinh(x: Double): Double = nativeSinh(x)\n\n**\n

```

\* Computes the hyperbolic cosine of the value [x].\n \* Special cases:\n \* - `cosh(NaN)` is `NaN`\n \* -  
`cosh(+Inf|-Inf)` is `+Inf`\n \* \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cosh(x: Double):

```

Double = nativeCosh(x)\n\n**\n * Computes the hyperbolic tangent of the value [x].\n * Special cases:\n * -
`tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n

```

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tanh(x: Double): Double =

```

nativeTanh(x)\n\n\*\*\n \* Computes the inverse hyperbolic sine of the value [x].\n \* The returned value is `y`  
such that `sinh(y) == x`.\n \* Special cases:\n \* - `asinh(NaN)` is `NaN`\n \* - `asinh(+Inf)` is `+Inf`\n \* -  
`asinh(-Inf)` is `-Inf`\n \* \n@SinceKotlin("1.2")\n@InlineOnly\npublic

```

actual inline fun asinh(x: Double): Double = nativeAsinh(x)\n\n**\n * Computes the inverse hyperbolic cosine of

```

the value [x].\n \* The returned value is positive `y` such that `cosh(y) == x`.\n \* Special cases:\n \* -  
`acosh(NaN)` is `NaN`\n \* - `acosh(x)` is `NaN` when `x < 1`\n \* - `acosh(+Inf)` is `+Inf`\n

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun acosh(x: Double): Double = nativeAcosh(x)\n\n**

```

```

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun acosh(x: Double): Double =
nativeAcosh(x)\n\n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n * The returned value is
`y` such that `tanh(y) == x`.\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x >
1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-1.0)` is `-Inf`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atanh(x: Double): Double =
nativeAtanh(x)\n\n/**\n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n * Special
cases:\n * - returns `+Inf` if
any of arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun hypot(x: Double, y: Double): Double =
nativeHypot(x, y)\n\n/**\n * Computes the positive square root of the value [x].\n * Special cases:\n * -
`sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
sqrt(x: Double): Double = nativeMath.sqrt(x)\n\n/**\n * Computes Euler's number `e` raised to the power of the
value [x].\n * Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun exp(x: Double): Double =
nativeMath.exp(x)\n\n/**\n * Computes `exp(x) - 1`.\n * This function can be implemented to produce more
precise result for [x] near zero.\n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * - `expm1(+Inf)` is `+Inf`\n
*\n
- `expm1(-Inf)` is `-1.0`\n
*\n @see [exp] function.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual
inline fun expm1(x: Double): Double = nativeExpml(x)\n\n/**\n * Computes the logarithm of the value [x] to the
given [base].\n * Special cases:\n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n * - `log(x, b)` is
`NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n * - `log(+Inf, +Inf)` is `NaN`\n * - `log(+Inf, b)` is `+Inf` for `b >
1` and `-Inf` for `b < 1`\n * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`\n * See also logarithm
functions for common fixed bases: [ln], [log10] and [log2].\n
*\n@SinceKotlin("1.2")\npublic actual fun log(x:
Double, base: Double): Double {\n if (base <= 0.0 || base == 1.0) return Double.NaN\n return nativeMath.log(x)
/ nativeMath.log(base)\n}\n\n/**\n * Computes the natural logarithm (base `E`) of the value [x].\n * Special
cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n
* - `ln(+Inf)` is `+Inf`\n * - `ln(0.0)` is `-Inf`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
ln(x: Double): Double = nativeMath.log(x)\n\n/**\n * Computes the common logarithm (base 10) of the value [x].\n
*\n @see [ln] function for special cases.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
log10(x: Double): Double = nativeLog10(x)\n\n/**\n * Computes the binary logarithm (base 2) of the value [x].\n
*\n @see [ln] function for special cases.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
log2(x: Double): Double = nativeLog2(x)\n\n/**\n * Computes `ln(x + 1)`.\n * This function can be
implemented to produce more precise result for [x] near zero.\n * Special cases:\n * - `ln1p(NaN)` is `NaN`\n *
- `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is `-Inf`\n * - `ln1p(+Inf)` is `+Inf`\n * @see [ln]
function\n * @see [expm1] function\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual
inline fun ln1p(x: Double): Double = nativeLog1p(x)\n\n/**\n * Rounds the given value [x] to an integer towards
positive infinity.\n * @return the smallest double value that is greater than or equal to the given value [x] and is a
mathematical integer.\n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a
mathematical integer.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun ceil(x: Double): Double =
nativeMath.ceil(x)\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.\n * @return the
largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.\n * Special
cases:\n * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun floor(x: Double): Double =
nativeMath.floor(x)\n\n/**\n * Rounds the given value [x] to an integer towards zero.\n
*\n @return the value [x] having its fractional part truncated.\n * Special cases:\n * - `truncate(x)` is `x`
where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun truncate(x: Double): Double =
nativeTrunc(x)\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even

```



integer.  
 \* Special cases: `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
`@SinceKotlin("1.2")`  
`public actual fun round(x: Double): Double {`  
 `if (x % 0.5 != 0.0) {`  
 `return nativeMath.round(x)`  
 `}`  
 `val floor = floor(x)`  
 `return if (floor % 2 == 0.0) floor else`  
 `ceil(x)`  
`}`  
 \* Returns the absolute value of the given value `[x]`.  
 \* Special cases: `abs(NaN)` is `NaN`.  
 \* @see `absoluteValue` extension property for `[Double]`  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun abs(x:`  
 `Double): Double = nativeMath.abs(x)`  
 \* Returns the sign of the given value `[x]`:  
 \* `-1.0` if the value is negative,  
 \* `0` if the value is zero,  
 \* `1.0` if the value is positive  
 \* Special case: `sign(NaN)` is `NaN`.  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun sign(x: Double): Double =`  
 `nativeSign(x)`  
 \* Returns the smaller of two values.  
 \* If either value is `NaN`, then the result is `NaN`.  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun min(a: Double, b: Double): Double =`  
 `nativeMath.min(a, b)`  
 \* Returns the greater of two values.  
 \* If either value is `NaN`, then the result is `NaN`.  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun max(a: Double, b: Double): Double =`  
 `nativeMath.max(a, b)`  
 \* Returns the cube root of `[x]`. For any `x`, `cbirt(-x) == -cbirt(x)`; that is, the cube root of a negative value is the negative of the cube root of that value's magnitude. Special cases:  
 \* Special cases:  
 \* If the argument is `NaN`, then the result is `NaN`.  
 \* If the argument is infinite, then the result is an infinity with the same sign as the argument.  
 \* If the argument is zero, then the result is a zero with the same sign as the argument.  
`@SinceKotlin("1.8")`  
`@WasExperimental(ExperimentalStdlibApi::class)`  
`@InlineOnly`  
`public actual inline fun cbirt(x: Double): Double = nativeMath.cbirt(x)`  
 // extensions  
 \* Raises this value to the power `[x]`.  
 \* Special cases:  
 \* `b.pow(0.0)` is `1.0`  
 \* `b.pow(1.0) == b`  
 \* `b.pow(NaN)` is `NaN`  
 \* `NaN.pow(x)` is `NaN` for `x != 0.0`  
 \* `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`  
 \* `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun Double.pow(x: Double): Double = nativeMath.pow(this, x)`  
 \* Raises this value to the integer power `[n]`.  
 \* See the other overload of `[pow]` for details.  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun Double.pow(n: Int): Double = nativeMath.pow(this, n.toDouble())`  
 \* Returns the absolute value of this value.  
 \* Special cases: `NaN.absoluteValue` is `NaN`.  
 \* @see `abs` function  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline val Double.absoluteValue: Double get() =`  
 `nativeMath.abs(this)`  
 \* Returns the sign of this value:  
 \* `-1.0` if the value is negative,  
 \* `0` if the value is zero,  
 \* `1.0` if the value is positive  
 \* Special case: `NaN.sign` is `NaN`  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline val Double.sign: Double get() =`  
 `nativeSign(this)`  
 \* Returns this value with the sign bit same as of the `[sign]` value.  
`@SinceKotlin("1.2")`  
`@InlineOnly`  
`public actual inline fun Double.withSign(sign: Int): Double =`  
 `this.withSign(sign.toDouble())`  
 \* Returns the ulp (unit in the last place) of this value.  
 \* An ulp is a positive distance between this value and the next nearest `[Double]` value larger in magnitude.  
 \* Special Cases:  
 \* `NaN.ulp` is `NaN`  
 \* `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf`  
 \* `0.0.ulp` is `Double.MIN_VALUE`  
`@SinceKotlin("1.2")`  
`public actual val Double.ulp: Double get() =`  
 `when {`  
 `this < 0 -> (-this).ulp`  
 `this.isNaN() || this == Double.POSITIVE_INFINITY -> this`  
 `this == Double.MAX_VALUE -> this - this.nextDown()`  
 `else -> this.nextUp() - this`  
 `}`  
 \* Returns the `[Double]` value nearest to this value in direction of positive infinity.  
`@SinceKotlin("1.2")`  
`public actual fun Double.nextUp(): Double =`  
 `when {`  
 `this.isNaN() || this == Double.POSITIVE_INFINITY -> this`  
 `this == 0.0 -> Double.MIN_VALUE`  
 `else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)`  
`}`  
 \* Returns the `[Double]` value nearest to this value in direction of negative infinity.  
`@SinceKotlin("1.2")`  
`public actual fun Double.nextDown(): Double =`  
 `when {`  
 `this.isNaN() || this == Double.NEGATIVE_INFINITY -> this`  
 `this == 0.0 -> -Double.MIN_VALUE`  
 `else -> Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)`  
`}`  
 \* Returns the `[Double]` value nearest to this value in direction from this value towards the value `[to]`.  
 \* Special cases:  
 \* `x.nextTowards(y)` is `NaN` if

either `x` or `y` are `NaN`  
`x.nextTowards(x) == x`  
`Double.nextTowards(to: Double): Double = when { this.isNaN() || to.isNaN() -> Double.NaN to == this -> to to > this -> this.nextUp() else /* to < this */ -> this.nextDown() }`  
Rounds this [Double] value to the nearest integer and converts the result to [Int].  
Ties are rounded towards positive infinity.  
Special cases:  
`x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`  
`x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`  
@throws `IllegalArgumentException` when this value is `NaN`  
`Double.roundToInt(): Int = when { this.isNaN() -> throw IllegalArgumentException("Cannot round NaN value.") this > Int.MAX_VALUE -> Int.MAX_VALUE this < Int.MIN_VALUE -> Int.MIN_VALUE else -> nativeMath.round(this).toInt() }`  
Rounds this [Double] value to the nearest integer and converts the result to [Long].  
Ties are rounded towards positive infinity.  
Special cases:  
`x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`  
`x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`  
@throws `IllegalArgumentException` when this value is `NaN`  
`Double.roundToLong(): Long = when { this.isNaN() -> throw IllegalArgumentException("Cannot round NaN value.") this > Long.MAX_VALUE -> Long.MAX_VALUE this < Long.MIN_VALUE -> Long.MIN_VALUE else -> nativeMath.round(this).toLong() }`  
region ===== Float Math  
===== Computes the sine of the angle [x] given in radians.  
Special cases:  
`sin(NaN|+Inf|-Inf)` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun sin(x: Float): Float = nativeMath.sin(x.toDouble()).toFloat()`  
Computes the cosine of the angle [x] given in radians.  
Special cases:  
`cos(NaN|+Inf|-Inf)` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun cos(x: Float): Float = nativeMath.cos(x.toDouble()).toFloat()`  
Computes the tangent of the angle [x] given in radians.  
Special cases:  
`tan(NaN|+Inf|-Inf)` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun tan(x: Float): Float = nativeMath.tan(x.toDouble()).toFloat()`  
Computes the arc sine of the value [x].  
\* the returned value is an angle in the range from `-PI/2` to `PI/2` radians.  
Special cases:  
`asin(x)` is `NaN`, when `abs(x) > 1` or `x` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun asin(x: Float): Float = nativeMath.asin(x.toDouble()).toFloat()`  
Computes the arc cosine of the value [x].  
\* the returned value is an angle in the range from `0.0` to `PI` radians.  
Special cases:  
`acos(x)` is `NaN`, when `abs(x) > 1` or `x` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun acos(x: Float): Float = nativeMath.acos(x.toDouble()).toFloat()`  
Computes the arc tangent of the value [x].  
\* the returned value is an angle in the range from `-PI/2` to `PI/2` radians.  
Special cases:  
`atan(NaN)` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun atan(x: Float): Float = nativeMath.atan(x.toDouble()).toFloat()`  
Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value `y / x`.  
\* the returned value is an angle in the range from `-PI` to `PI` radians.  
Special cases:  
`atan2(0.0, 0.0)` is `0.0`  
`atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`  
`atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`  
`atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for `-Inf < y < 0`  
`atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`  
`atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`  
`atan2(+Inf, x)` is `PI/2` for finite `x`  
`atan2(-Inf, x)` is `-PI/2` for finite `x`  
`atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun atan2(y: Float, x: Float): Float = nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()`  
Computes the hyperbolic sine of the value [x].  
Special cases:  
`sinh(NaN)` is `NaN`  
`sinh(+Inf)` is `+Inf`  
`sinh(-Inf)` is `-Inf`  
`@SinceKotlin("1.2") @InlineOnly public actual inline fun sinh(x: Float): Float = nativeSinh(x.toDouble()).toFloat()`  
Computes the hyperbolic cosine of the value [x].  
Special

cases:  $\cosh(\text{NaN})$  is  $\text{NaN}$   $\cosh(+\text{Inf}-\text{Inf})$  is  $+\text{Inf}$

```
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cosh(x: Float): Float =
nativeCosh(x.toDouble()).toFloat()\n\n**\n * Computes the hyperbolic tangent of the value [x].\n * Special
cases:\n * - \tanh(\text{NaN}) is \text{NaN}\n * - \tanh(+\text{Inf}) is 1.0\n * - \tanh(-\text{Inf}) is -1.0\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tanh(x: Float): Float =
nativeTanh(x.toDouble()).toFloat()\n\n**\n * Computes the inverse hyperbolic sine of the value [x].\n * The
returned value is `y` such that \sinh(y) == x`.\n * Special cases:\n * - \asinh(\text{NaN})
is \text{NaN}\n * - \asinh(+\text{Inf}) is +\text{Inf}\n * - \asinh(-\text{Inf}) is -\text{Inf}\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun asinh(x: Float): Float =
nativeAsinh(x.toDouble()).toFloat()\n\n**\n * Computes the inverse hyperbolic cosine of the value [x].\n * The
returned value is positive `y` such that \cosh(y) == x`.\n * Special cases:\n * - \acosh(\text{NaN}) is \text{NaN}\n * -
\acosh(x) is \text{NaN} when `x < 1`\n * - \acosh(+\text{Inf}) is +\text{Inf}\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic
actual inline fun acosh(x: Float): Float = nativeAcosh(x.toDouble()).toFloat()\n\n**\n * Computes the inverse
hyperbolic tangent of the value [x].\n * The returned value is `y` such that \tanh(y) == x`.\n * Special
cases:\n * - \tanh(\text{NaN}) is \text{NaN}\n * - \tanh(x) is \text{NaN} when `x > 1` or `x < -1`\n * - \tanh(1.0) is +\text{Inf}\n *
- \tanh(-1.0) is -\text{Inf}\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atanh(x: Float): Float =
nativeAtanh(x.toDouble()).toFloat()\n\n**\n * Computes \sqrt(x^2 + y^2) without intermediate overflow or underflow.\n * Special cases:\n * - returns
+\text{Inf} if any of arguments is infinite\n * - returns \text{NaN} if any of arguments is \text{NaN} and the other is not infinite\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun hypot(x: Float, y: Float): Float =
nativeHypot(x.toDouble(), y.toDouble()).toFloat()\n\n**\n * Computes the positive square root of the value [x].\n
*\n * Special cases:\n * - \sqrt(x) is \text{NaN} when `x < 0` or `x` is \text{NaN}\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sqrt(x: Float): Float =
nativeMath.sqrt(x.toDouble()).toFloat()\n\n**\n * Computes Euler's number `e` raised to the power of the value
[x].\n * Special cases:\n * - \exp(\text{NaN}) is \text{NaN}\n * - \exp(+\text{Inf}) is +\text{Inf}\n * - \exp(-\text{Inf}) is 0.0\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun exp(x: Float): Float =
nativeMath.exp(x.toDouble()).toFloat()\n\n**\n * Computes \exp(x) - 1`.\n * This function can be implemented to produce more precise result for [x] near
zero.\n * Special cases:\n * - \expm1(\text{NaN}) is \text{NaN}\n * - \expm1(+\text{Inf}) is +\text{Inf}\n * - \expm1(-\text{Inf}) is -
1.0\n * * @see [exp] function.\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun expm1(x:
Float): Float = nativeExp1(x.toDouble()).toFloat()\n\n**\n * Computes the logarithm of the value [x] to the given
[base].\n * Special cases:\n * - \log(x, b) is \text{NaN} if either `x` or `b` are \text{NaN}\n * - \log(x, b) is \text{NaN}
when `x < 0` or `b <= 0` or `b == 1.0`\n * - \log(+\text{Inf}, +\text{Inf}) is \text{NaN}\n * - \log(+\text{Inf}, b) is +\text{Inf} for `b > 1` and
-\text{Inf} for `b < 1`\n * - \log(0.0, b) is -\text{Inf} for `b > 1` and +\text{Inf} for `b > 1`\n * * See also logarithm functions
for common fixed bases: [ln], [log10] and [log2].\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline
fun log(x: Float, base: Float): Float = log(x.toDouble(), base.toDouble()).toFloat()\n\n**\n * Computes the natural logarithm (base `E`) of the value [x].\n * Special cases:\n * - \ln(\text{NaN}) is \text{NaN}\n * -
\ln(x) is \text{NaN} when `x < 0.0`\n * - \ln(+\text{Inf}) is +\text{Inf}\n * - \ln(0.0) is -\text{Inf}\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun ln(x: Float): Float =
nativeMath.log(x.toDouble()).toFloat()\n\n**\n * Computes the common logarithm (base 10) of the value [x].\n *
*\n * @see [ln] function for special cases.\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun log10(x:
Float): Float = nativeLog10(x.toDouble()).toFloat()\n\n**\n * Computes the binary logarithm (base 2) of the value
[x].\n * * @see [ln] function for special cases.\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline
fun log2(x: Float): Float = nativeLog2(x.toDouble()).toFloat()\n\n**\n * Computes \ln(x + 1)`. \n * This
function can be implemented to produce more precise result for [x] near zero.\n * Special cases:\n
* - \ln1p(\text{NaN}) is \text{NaN}\n * - \ln1p(x) is \text{NaN} where `x < -1.0`\n * - \ln1p(-1.0) is -\text{Inf}\n * - \ln1p(+\text{Inf})
is +\text{Inf}\n * * @see [ln] function\n * * @see [expm1] function\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic
actual inline fun ln1p(x: Float): Float = nativeLog1p(x.toDouble()).toFloat()\n\n**\n * Rounds the given value [x]
```

to an integer towards positive infinity.  
`@return the smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.`  
`* Special cases:`  
`- `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun ceil(x: Float): Float = nativeMath.ceil(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.`  
`@return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.`  
`* Special cases:`  
`- `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun floor(x: Float): Float = nativeMath.floor(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] to an integer towards zero.`  
`@return the value [x] having its fractional part truncated.`  
`* Special cases:`  
`- `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun truncate(x: Float): Float = truncate(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.`  
`* Special cases:`  
`- `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun round(x: Float): Float = round(x.toDouble()).toFloat()\n\n/**\n * Returns the absolute value of the given value [x].`  
`* Special cases:`  
`- `abs(NaN)` is `NaN``  
`* @see absoluteValue extension property for [Float]`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun abs(x: Float): Float = nativeMath.abs(x.toDouble()).toFloat()\n\n/**\n * Returns the sign of the given value [x]:`  
`- `-1.0` if the value is negative,`  
`- zero if the value is zero,`  
`- `1.0` if the value is positive`  
`* Special case:`  
`- `sign(NaN)` is `NaN``  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sign(x: Float): Float = nativeSign(x.toDouble()).toFloat()\n\n/**\n * Returns the smaller of two values.`  
`* If either value is `NaN`, then the result is `NaN`.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun min(a: Float, b: Float): Float = nativeMath.min(a, b)\n\n/**\n * Returns the greater of two values.`  
`* If either value is `NaN`, then the result is `NaN`.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun max(a: Float, b: Float): Float = nativeMath.max(a, b)\n\n/**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`; that is, the cube root of a negative value is the negative of the cube root of that value's magnitude. Special cases:`  
`* Special cases:`  
`- If the argument is `NaN`, then the result is `NaN`.`  
`- If the argument is infinite, then the result is an infinity with the same sign as the argument.`  
`- If the argument is zero, then the result is a zero with the same sign as the argument.`  
`*\n@SinceKotlin("1.8")\n@WasExperimental(ExperimentalStdlibApi::class)\n@InlineOnly\npublic actual inline fun cbrt(x: Float): Float = nativeMath.cbrt(x.toDouble()).toFloat()\n\n// extensions\n\n/**\n * Raises this value to the power [x].`  
`* Special cases:`  
`- `b.pow(0.0)` is `1.0``  
`- `b.pow(1.0) == b``  
`- `b.pow(NaN)` is `NaN``  
`- `NaN.pow(x)` is `NaN` for `x != 0.0``  
`- `b.pow(Inf)` is `NaN` for `abs(b) == 1.0``  
`- `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.pow(x: Float): Float = nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()\n\n/**\n * Raises this value to the integer power [n].`  
`* See the other overload of [pow] for details.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.pow(n: Int): Float = nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()\n\n/**\n * Returns the absolute value of this value.`  
`* Special cases:`  
`- `NaN.absoluteValue` is `NaN``  
`* @see abs function`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Float.absoluteValue: Float get() = nativeMath.abs(this.toDouble()).toFloat()\n\n/**\n * Returns the sign of this value:`  
`- `-1.0` if the value is negative,`  
`- zero if the value is zero,`  
`- `1.0` if the value is positive`  
`* Special case:`  
`- `NaN.sign` is `NaN``  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Float.sign: Float get() = nativeSign(this.toDouble()).toFloat()\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.`  
`* If [sign] is `NaN` the sign of the result is undefined.`  
`*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Float): Float = this.toDouble().withSign(sign.toDouble()).toFloat()\n\n/**\n * Returns this value with the sign bit same as of the`

```

[sign] value.\n *^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Int): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()\n\n/**\n * Rounds this [Float] value to the nearest integer and
converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * -
`x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE`
when `x < Int.MIN_VALUE`\n * * @throws IllegalArgumentException when this value
is `NaN`\n *^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToInt(): Int =
toDouble().roundToInt()\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to
[Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `x.roundToLong() ==
Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x
< Long.MIN_VALUE`\n * * @throws IllegalArgumentException when this value is `NaN`\n
*^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToLong(): Long =
toDouble().roundToLong()\n\n// endregion\n\n// region ===== Integer Math
=====
\n\n/**\n * Returns the absolute value of the given value
[n].\n * Special cases:\n * - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow\n * * @see
absoluteValue extension property for [Int]\n *^n// TODO: remove manual 'or' when KT-19290 is
fixed\n@SinceKotlin("1.2")\npublic
actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n\n\n/**\n * Returns the smaller of two values.\n
*^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun min(a: Int, b: Int): Int = nativeMath.min(a,
b)\n\n/**\n * Returns the greater of two values.\n *^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
max(a: Int, b: Int): Int = nativeMath.max(a, b)\n\n/**\n * Returns the absolute value of this value.\n * Special
cases:\n * - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n * * @see abs
function\n *^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Int.absoluteValue: Int get() =
abs(this)\n\n/**\n * Returns the sign of this value:\n * - `-1` if the value is negative,\n * - `0` if the value is zero,\n
* - `1` if the value is positive\n *^n@SinceKotlin("1.2")\npublic actual val Int.sign: Int get() = when {\n this <
0 -> -1\n this > 0 -> 1\n else -> 0}\n\n/**\n * Returns the
absolute value of the given value [n].\n * Special cases:\n * - `abs(Long.MIN_VALUE)` is
`Long.MIN_VALUE` due to an overflow\n * * @see absoluteValue extension property for [Long]\n
*^n@SinceKotlin("1.2")\npublic actual fun abs(n: Long): Long = if (n < 0) -n else n\n\n/**\n * Returns the smaller
of two values.\n *^n@SinceKotlin("1.2")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun
min(a: Long, b: Long): Long = if (a <= b) a else b\n\n/**\n * Returns the greater of two values.\n
*^n@SinceKotlin("1.2")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun max(a: Long, b:
Long): Long = if (a >= b) a else b\n\n/**\n * Returns the absolute value of this value.\n * Special cases:\n * -
`Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n * * @see abs function\n
*^n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Long.absoluteValue: Long get() =
abs(this)\n\n/**\n * Returns the sign of this value:\n * - `-1` if the
value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n
*^n@SinceKotlin("1.2")\npublic actual val Long.sign: Int get() = when {\n this < 0 -> -1\n this > 0 -> 1\n
else -> 0}\n\n// endregion\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin\n\n/**\n * Returns `true` if the specified number is a\n * Not-a-
Number (NaN) value, `false` otherwise.\n *^npublic actual fun Double.isNaN(): Boolean = this != this\n\n/**\n *
Returns `true` if the specified number is a\n * Not-a-Number (NaN) value, `false` otherwise.\n *^npublic actual fun
Float.isNaN(): Boolean = this != this\n\n/**\n * Returns `true` if this value is infinitely large in magnitude.\n
*^npublic actual fun Double.isInfinite(): Boolean = this == Double.POSITIVE_INFINITY || this ==
Double.NEGATIVE_INFINITY\n\n/**\n * Returns `true` if this value is infinitely large in magnitude.\n *^npublic actual fun Float.isInfinite(): Boolean =
this == Float.POSITIVE_INFINITY || this == Float.NEGATIVE_INFINITY\n\n/**\n * Returns `true` if the
argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).\n *^npublic

```

actual fun Double.isFinite(): Boolean = !isInfinite() && !isNaN()\n\n/\*\*\n \* Returns `true` if the argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).\n \*/\npublic actual fun Float.isFinite(): Boolean = !isInfinite() && !isNaN()\n\n/\*\*\n \* Counts the number of set bits in the binary representation of this [Int] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.countOneBits(): Int {\n // Hacker's Delight 5-1 algorithm\n var v = this\n v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)\n v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)\n v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)\n v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)\n v = (v and 0x0000FFFF) + (v.ushr(16))\n return v}\n\n/\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Int.countLeadingZeroBits(): Int = nativeClz32(this)\n\n/\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.countTrailingZeroBits(): Int =\n // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with countLeadingZeroBits\n Int.SIZE\_BITS - (this or -this).inv().countLeadingZeroBits()\n\n/\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [Int] number, or zero, if this number is zero.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.takeHighestOneBit(): Int =\n if (this == 0) 0 else 1.shl(Int.SIZE\_BITS - 1 - countLeadingZeroBits())\n\n/\*\*\n \* Returns a number having a single bit set in the position of the least significant set bit of this [Int] number, or zero, if this number is zero.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.takeLowestOneBit(): Int =\n // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit\n this and -this\n\n/\*\*\n \* Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.\n \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n \*/\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.rotateLeft(bitCount: Int): Int =\n shl(bitCount) or ushr(Int.SIZE\_BITS - bitCount)\n\n/\*\*\n \* Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.\n \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n \* `number.rotateRight(-n) == number.rotateLeft(n)`\n \* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally\n \* `number.rotateRight(n) == number.rotateRight(n % 32)`\n \*/\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.rotateRight(bitCount: Int): Int =\n shl(Int.SIZE\_BITS - bitCount) or ushr(bitCount)\n\n/\*\*\n \* Counts the number of set bits in the binary representation of this [Long] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.countOneBits(): Int =\n high.countOneBits() + low.countOneBits()\n\n/\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.countLeadingZeroBits(): Int =\n when (val high = this.high) {\n 0 -> Int.SIZE\_BITS + low.countLeadingZeroBits()\n else -> high.countLeadingZeroBits()\n }\n\n/\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n \*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic



```

the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\nimport kotlin.math.pow\n\ninternal actual fun
defaultPlatformRandom(): Random =\n Random(js(\n (Math.random() * Math.pow(2, 32)) |
0\").unsafeCast<Int>())\n\nprivate val INV_2_26: Double = 2.0.pow(-26)\nprivate val INV_2_53: Double =
2.0.pow(-53)\n\ninternal actual fun doubleFromParts(hi26: Int, low27: Int): Double =\n hi26 * INV_2_26 + low27 *
INV_2_53\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.reflect\n\nimport findAssociatedObject\n\n/**\n * The experimental marker for associated
objects API.\n * Any usage of a declaration annotated with `@ExperimentalAssociatedObjects` must be
accepted either by\n * annotating that usage with the [OptIn] annotation, e.g.
`@OptIn(ExperimentalAssociatedObjects::class)`,\n * or by using the compiler
argument `-opt-in=kotlin.reflect.ExperimentalAssociatedObjects`.\n */\n\n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n@Retention(value = AnnotationRetention.BINARY)\n\npublic annotation class
ExperimentalAssociatedObjects\n\n/**\n * Makes the annotated annotation class an associated object key.\n */\n
An associated object key annotation should have single [KClass] parameter.\n * When applied to a class with
reference to an object declaration as an argument, it binds\n * the object to the class, making this binding
discoverable at runtime using [findAssociatedObject].\n
*/\n\n@ExperimentalAssociatedObjects\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.A
NNOTATION_CLASS)\n\npublic annotation class AssociatedObjectKey\n\n/**\n * If [T] is an
@[AssociatedObjectKey]-annotated annotation class and [this] class is annotated with @[T] (`S::class`),\n * returns
object `S`.\n * Otherwise returns `null`.\n */\n\n@ExperimentalAssociatedObjects\n\npublic inline fun <reified
T : Annotation> KClass<*>.findAssociatedObject(): Any? =\n this.findAssociatedObject(T::class)\n\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\nimport getKClass\n\nimport kotlin.reflect.KClass\n\nimport kotlin.reflect.js.internal.KClassImpl\n\n/**\n
 * Represents the constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a
constructor reference.\n */\n\nexternal interface JsClass<T : Any> {\n /**\n * Returns the unqualified name of the
class represented by this instance.\n */\n val name: String\n}\n\n/**\n * Obtains a constructor reference for the
given `KClass`.\n */\n\nval <T : Any> KClass<T>.js: JsClass<T>\n get() = (this as KClassImpl<T>).jClass\n\n/**\n
 * Obtains a `KClass` instance for the given constructor reference.\n */\n\nval <T : Any> JsClass<T>.kotlin:
KClass<T>\n get() = getKClass(this)\n\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal abstract
class KClassImpl<T : Any>(\n internal open val jClass: JsClass<T>\n) : KClass<T> {\n\n override val
qualifiedName: String?\n get() = TODO()\n\n override fun equals(other: Any?): Boolean {\n return other
is KClassImpl<*> && jClass == other.jClass\n }\n\n // TODO: use FQN\n override fun hashCode(): Int =
simpleName?.hashCode() ?: 0\n\n override fun toString(): String {\n // TODO: use FQN\n return \"class
$simpleName\"\n }\n}\n\ninternal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) :
KClassImpl<T>(jClass) {\n override val simpleName: String? =
jClass.asDynamic().`$metadata$`?.simpleName.unsafeCast<String?>()\n\n override fun isInstance(value: Any?): Boolean {\n
 return jsIsType(value, jClass)\n }\n}\n\ninternal class
PrimitiveKClassImpl<T : Any>(\n jClass: JsClass<T>,\n private val givenSimpleName: String,\n private val
isInstanceFunction: (Any?) -> Boolean\n) : KClassImpl<T>(jClass) {\n override fun equals(other: Any?): Boolean
{\n if (other !is PrimitiveKClassImpl<*>) return false\n return super.equals(other) && givenSimpleName
== other.givenSimpleName\n }\n\n override val simpleName: String? get() = givenSimpleName\n\n override
fun isInstance(value: Any?): Boolean {\n return isInstanceFunction(value)\n }\n}\n\ninternal object
NothingKClassImpl : KClassImpl<Nothing>(js(\"Object\")) {\n override val simpleName: String =
\"Nothing\"\n\n override fun isInstance(value: Any?): Boolean = false\n\n override val jClass:
JsClass<Nothing>\n get() = throw UnsupportedOperationException(\"There's no native

```



```

JS class for Nothing type`)
 override fun equals(other: Any?): Boolean = other === this
 override fun hashCode(): Int = 0
}

internal class ErrorKClass : KClass<Nothing> {
 override val simpleName: String?
 get() = error("Unknown simpleName for ErrorKClass")
 override val qualifiedName: String?
 get() = error("Unknown qualifiedName for ErrorKClass")
 override fun isInstance(value: Any?): Boolean =
 error("Can's check isInstance on ErrorKClass")
 override fun equals(other: Any?): Boolean = other ===
 this
 override fun hashCode(): Int = 0
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.
 */
package kotlin.reflect

internal actual inline val
KClass<*>.qualifiedOrSimpleName: String?
 get() = simpleName

/*
 * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language
contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
 */
package // a package is omitted to get declarations directly under the module
// TODO:
Remove once JsReflectionAPICallChecker supports more reflection
types

@file:Suppress("Unsupported")
import kotlin.reflect.*
import
kotlin.reflect.js.internal.*

@JsName("createKType")
internal fun createKType(
 classifier: KClassifier,
 arguments: Array<KTypeProjection>,
 isMarkedNullable: Boolean
) =
 KTypeImpl(classifier,
arguments.asList(), isMarkedNullable)

@JsName("createDynamicKType")
internal fun
createDynamicKType(): KType = DynamicKType

@JsName("markKTypeNullable")
internal fun
markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments,
true)

@JsName("createKTypeParameter")
internal fun createKTypeParameter(
 name: String,
 upperBounds: Array<KType>,
 variance: String
): KTypeParameter
{
 val kVariance = when (variance) {
 "in" -> KVariance.IN
 "out" -> KVariance.OUT
 else -
> KVariance.INVARIANT
 }
 return KTypeParameterImpl(name, upperBounds.asList(), kVariance,
false)
}

@JsName("getStarKTypeProjection")
internal fun getStarKTypeProjection(): KTypeProjection =
 KTypeProjection.STAR

@JsName("createCovariantKTypeProjection")
internal fun
createCovariantKTypeProjection(type: KType): KTypeProjection =
 KTypeProjection.covariant(type)

@JsName("createInvariantKTypeProjection")
internal fun
createInvariantKTypeProjection(type: KType): KTypeProjection =
 KTypeProjection.invariant(type)

@JsName("createContravariantKTypeProjection")
internal fun
createContravariantKTypeProjection(type: KType): KTypeProjection =
 KTypeProjection.contravariant(type)

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.
 * Use of this source code is governed by
the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package
kotlin.reflect.js.internal
import kotlin.reflect.*

internal class KTypeImpl(
 override val classifier:
KClassifier,
 override val arguments: List<KTypeProjection>,
 override val isMarkedNullable: Boolean
):
 KType {
 override fun equals(other: Any?): Boolean =
 other is KTypeImpl &&
 classifier ==
other.classifier && arguments == other.arguments && isMarkedNullable == other.isMarkedNullable
 override
fun hashCode(): Int =
 (classifier.hashCode() * 31 + arguments.hashCode()) * 31 +
isMarkedNullable.hashCode()
 override fun toString(): String {
 val kClass = (classifier as? KClass<*>)
 val classifierName = when {
 kClass == null -> classifier.toString()
 kClass.simpleName != null
-> kClass.simpleName
 else -> "(non-denotable type)"
 }
 val args =
 if
(arguments.isEmpty()) ""
 else arguments.joinToString(", ", "<", ">")
 val nullable = if
(isMarkedNullable) "?"
 else ""
 return classifierName + args + nullable
 }
}

internal object
DynamicKType : KType {
 override val classifier: KClassifier? = null
 override val arguments:
List<KTypeProjection> = emptyList()
 override val isMarkedNullable: Boolean = false
 override fun
toString(): String = "dynamic"

}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
 */
package kotlin.reflect.js.internal
import kotlin.reflect.*

internal data class

```

```

KTypeParameterImpl(\n override val name: String,\n override val upperBounds: List<KType>,\n override val variance: KVariance,\n override val isReified: Boolean)\n) : KTypeParameter {\n override fun toString(): String = name\n}, "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.js.JsClass\n\n@JsName("PrimitiveClasses")\n\ninternal object PrimitiveClasses {\n @JsName("anyClass")\n val anyClass = PrimitiveKClassImpl(js("Object").unsafeCast<JsClass<Any>>(), "Any", { it is Any })\n\n @JsName("numberClass")\n val numberClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Number>>(), "Number", { it is Number })\n\n @JsName("nothingClass")\n val nothingClass = NothingKClassImpl\n\n @JsName("booleanClass")\n val booleanClass = PrimitiveKClassImpl(js("Boolean").unsafeCast<JsClass<Boolean>>(), "Boolean", { it is Boolean })\n\n @JsName("byteClass")\n val byteClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Byte>>(), "Byte", { it is Byte })\n\n @JsName("shortClass")\n val shortClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Short>>(), "Short", { it is Short })\n\n @JsName("intClass")\n val intClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(), "Int", { it is Int })\n\n @JsName("floatClass")\n val floatClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(), "Float", { it is Float })\n\n @JsName("doubleClass")\n val doubleClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(), "Double", { it is Double })\n\n @JsName("arrayClass")\n val arrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>>(), "Array", { it is Array<*> })\n\n @JsName("stringClass")\n val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(), "String", { it is String })\n\n @JsName("throwableClass")\n val throwableClass = PrimitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable })\n\n @JsName("booleanArrayClass")\n val booleanArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<BooleanArray>>(), "BooleanArray", { it is BooleanArray })\n\n @JsName("charArrayClass")\n val charArrayClass = PrimitiveKClassImpl(js("Uint16Array").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray })\n\n @JsName("byteArrayClass")\n val byteArrayClass = PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray })\n\n @JsName("shortArrayClass")\n val shortArrayClass = PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray })\n\n @JsName("intArrayClass")\n val intArrayClass = PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntArray>>(), "IntArray", { it is IntArray })\n\n @JsName("longArrayClass")\n val longArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray })\n\n @JsName("floatArrayClass")\n val floatArrayClass = PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray })\n\n @JsName("doubleArrayClass")\n val doubleArrayClass = PrimitiveKClassImpl(js("Float64Array").unsafeCast<JsClass<DoubleArray>>(), "DoubleArray", { it is DoubleArray })\n\n @JsName("functionClass")\n fun functionClass(arity: Int): KClassImpl<Any> {\n return functionClasses.get(arity) ?: run {\n val result = PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function$arity", {\n { jsTypeOf(it) === "function" && it.asDynamic().length === arity })\n }\n functionClasses.asDynamic()[arity] = result\n result\n }\n }\n\n private val functionClasses = arrayOfNulls<KClassImpl<Any>>(0), "/*\n *

```

```

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is
omitted to get declarations directly under the module\n\nimport kotlin.reflect.*\nimport
kotlin.reflect.js.internal.*\n\n@JsName("\ngetKClass")\ninternal fun <T : Any> getKClass(jClass: Any /*
JsClass<T> | Array<JsClass<T>> */): KClass<T> {\n return if (js("\nArray").isArray(jClass)) {\n
getKClassM(jClass.unsafeCast<Array<JsClass<T>>())\n } else {\n
getKClass1(jClass.unsafeCast<JsClass<T>>())\n }\n}\n\n@JsName("\ngetKClassM")\ninternal fun <T : Any>
getKClassM(jClasses: Array<JsClass<T>>): KClass<T> = when (jClasses.size) {\n 1 ->
getKClass1(jClasses[0])\n 0 -> Nothing\n KClassImpl.unsafeCast<KClass<T>>()\n else ->
ErrorKClass().unsafeCast<KClass<T>>()\n}\n\n@JsName("\ngetKClassFromExpression")\ninternal fun <T : Any>
getKClassFromExpression(e:
T): KClass<T> =\n when (jsTypeOf(e)) {\n "string" -> PrimitiveClasses.stringClass\n "number" -> if
(jsBitwiseOr(e, 0).asDynamic() === e) PrimitiveClasses.intClass else PrimitiveClasses.doubleClass\n
 "boolean" -> PrimitiveClasses.booleanClass\n "function" ->
PrimitiveClasses.functionClass(e.asDynamic().length)\n else -> {\n when {\n e is BooleanArray
-> PrimitiveClasses.booleanArrayClass\n e is CharArray -> PrimitiveClasses.charArrayClass\n
 e is ByteArray -> PrimitiveClasses.byteArrayClass\n e is ShortArray -> PrimitiveClasses.shortArrayClass\n
 e is IntArray -> PrimitiveClasses.intArrayClass\n e is LongArray ->
PrimitiveClasses.longArrayClass\n e is FloatArray -> PrimitiveClasses.floatArrayClass\n e is
DoubleArray -> PrimitiveClasses.doubleArrayClass\n e is KClass<*> -> KClass::class\n
 e is Array<*> -> PrimitiveClasses.arrayClass\n else -> {\n val constructor =
js("\nObject").getPrototypeOf(e).constructor\n when {\n constructor === js("\nObject") ->
PrimitiveClasses.anyClass\n constructor === js("\nError") -> PrimitiveClasses.throwableClass\n
 else -> {\n val jsClass: JsClass<T> = constructor\n getKClass1(jsClass)\n
 }\n }\n }\n }\n }\n }\n}\n\nunsafeCast<KClass<T>>()\n\n@JsName("\ngetKClass1")\ninternal fun <T : Any> getKClass1(jClass:
JsClass<T>): KClass<T> {\n if (jClass === js("\nString")) return
PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()\n val metadata = jClass.asDynamic().`$metadata$\n\n
return if (metadata != null) {\n if (metadata.`$kClass$` == null) {\n val kClass =
SimpleKClassImpl(jClass)\n metadata.`$kClass$` = kClass\n kClass\n } else {\n
metadata.`$kClass$`\n }\n } else {\n SimpleKClassImpl(jClass)\n }\n}\n\n"/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Exposes
the JavaScript [RegExp
object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/RegExp) to Kotlin.\n
*/\n\n@Suppress("\nNOT_DOCUMENTED")\npublic external class RegExp(pattern: String, flags: String? =
definedExternally) {\n public fun test(str: String): Boolean\n public fun exec(str: String): RegExpMatch?\n\n
public override fun toString(): String\n\n /**\n * The lastIndex is a read/write integer property of regular
expressions that specifies the index at which to start the next match.\n
*/\n public var lastIndex: Int\n\n public val global: Boolean\n public val ignoreCase: Boolean\n public val
multiline: Boolean\n}\n\n/**\n * Resets the regular expression so that subsequent [RegExp.test] and [RegExp.exec]
calls will match starting with the beginning of the input string.\n */\n\npublic fun RegExp.reset() {\n lastIndex =
0\n}\n\n// TODO: Inherit from array or introduce asArray() extension\n\n/**\n * Represents the return value of
[RegExp.exec].\n */\n\n@Suppress("\nNOT_DOCUMENTED")\npublic external interface RegExpMatch {\n public
val index: Int\n public val input: String\n public val length: Int\n}\n\n/**\n * Returns the entire text matched by
[RegExp.exec] if the [index] parameter is 0, or the text matched by the capturing parenthesis\n * at the given
index.\n */\n\npublic inline operator fun RegExpMatch.get(index: Int): String? = asDynamic()[index]\n\n/**\n *
Converts the result of [RegExp.exec] to an array where the first element contains the entire

```

```

matched text and each subsequent\n * element is the text matched by each capturing parenthesis.\n *\npublic inline
fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String?>>()\n", /*\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\npackage kotlin.sequences\n\ninternal
actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {\n private
var sequenceRef: Sequence<T>? = sequence\n\n actual override fun iterator(): Iterator<T> {\n val sequence =
sequenceRef ?: throw IllegalStateException("This sequence can be consumed only once.")\n sequenceRef =
null\n return sequence.iterator()\n }\n}\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *\n\npackage
kotlin.text\n\n@SinceKotlin("1.5")\npublic actual enum class CharCategory(internal val value: Int, public actual
val code: String) {\n /**\n * General category \"Cn\" in the Unicode specification.\n * UNASSIGNED(0,
\"Cn\"),\n\n /**\n * General category \"Lu\" in the Unicode specification.\n * UPPERCASE_LETTER(1,
\"Lu\"),\n\n /**\n * General category \"Ll\" in the Unicode specification.\n * LOWERCASE_LETTER(2,
\"Ll\"),\n\n /**\n * General category \"Lt\" in the Unicode specification.\n * TITLECASE_LETTER(3,
\"Lt\"),\n\n /**\n * General category \"Lm\" in the Unicode specification.\n * MODIFIER_LETTER(4,
\"Lm\"),\n\n /**\n * General category \"Lo\" in the Unicode specification.\n * OTHER_LETTER(5,
\"Lo\"),\n\n /**\n * General category \"Mn\" in the Unicode specification.\n * NON_SPACING_MARK(6,
\"Mn\"),\n\n /**\n * General category \"Me\" in the Unicode specification.\n * ENCLOSING_MARK(7,
\"Me\"),\n\n /**\n * General category \"Mc\" in the Unicode specification.\n * COMBINING_SPACING_MARK(8,
\"Mc\"),\n\n /**\n * General category \"Nd\" in the Unicode
specification.\n * DECIMAL_DIGIT_NUMBER(9, \"Nd\"),\n\n /**\n * General category \"NI\" in the
Unicode specification.\n * LETTER_NUMBER(10, \"NI\"),\n\n /**\n * General category \"No\" in the
Unicode specification.\n * OTHER_NUMBER(11, \"No\"),\n\n /**\n * General category \"Zs\" in the
Unicode specification.\n * SPACE_SEPARATOR(12, \"Zs\"),\n\n /**\n * General category \"Zl\" in the
Unicode specification.\n * LINE_SEPARATOR(13, \"Zl\"),\n\n /**\n * General category \"Zp\" in the
Unicode specification.\n * PARAGRAPH_SEPARATOR(14, \"Zp\"),\n\n /**\n * General category
\"Cc\" in the Unicode
specification.\n * CONTROL(15, \"Cc\"),\n\n /**\n * General category \"Cf\" in the Unicode
specification.\n * FORMAT(16, \"Cf\"),\n\n /**\n * General category \"Co\" in the Unicode
specification.\n * PRIVATE_USE(18, \"Co\"),\n\n /**\n * General category \"Cs\" in the Unicode
specification.\n * SURROGATE(19, \"Cs\"),\n\n /**\n * General category \"Pd\" in the Unicode
specification.\n * DASH_PUNCTUATION(20, \"Pd\"),\n\n /**\n * General category \"Ps\" in the
Unicode specification.\n * START_PUNCTUATION(21, \"Ps\"),\n\n /**\n * General category \"Pe\" in
the Unicode specification.\n * END_PUNCTUATION(22, \"Pe\"),\n\n /**\n * General category \"Pc\" in
the Unicode specification.\n * CONNECTOR_PUNCTUATION(23, \"Pc\"),\n\n /**\n * General
category \"Po\" in the Unicode specification.\n * OTHER_PUNCTUATION(24, \"Po\"),\n\n /**\n *
General
category \"Sm\" in the Unicode specification.\n * MATH_SYMBOL(25, \"Sm\"),\n\n /**\n * General
category \"Sc\" in the Unicode specification.\n * CURRENCY_SYMBOL(26, \"Sc\"),\n\n /**\n *
General category \"Sk\" in the Unicode specification.\n * MODIFIER_SYMBOL(27, \"Sk\"),\n\n /**\n *
General category \"So\" in the Unicode specification.\n * OTHER_SYMBOL(28, \"So\"),\n\n /**\n *
General category \"Pi\" in the Unicode specification.\n * INITIAL_QUOTE_PUNCTUATION(29,
\"Pi\"),\n\n /**\n * General category \"Pf\" in the Unicode specification.\n * FINAL_QUOTE_PUNCTUATION(30, \"Pf\");\n\n /**\n * Returns `true` if [char] character belongs to this
category.\n * public actual operator fun contains(char: Char): Boolean = char.getCategoryValue() ==
this.value\n\n companion object {\n internal fun valueOf(category: Int): CharCategory =\n when

```

```

(category) {\n
 in 0..16 -> values()[category]\n in 18..30 -> values()[category - 1]\n else -> throw
IllegalArgumentException("Category #${category} is not defined.")\n }\n }\n\n", /*\n * Copyright 2010-
2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * The
exception thrown when a character encoding or decoding error occurs.\n
*\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual open class
CharacterCodingException(message: String?) : Exception(message) {\n actual constructor() : this(null)\n }\n\n", /*\n
* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.text\n\n/**\n * A mutable
sequence of characters.\n *\n * String builder can be used to efficiently perform multiple string manipulation
operations.\n *\n public actual class StringBuilder actual constructor(content: String) : Appendable, CharSequence
{\n /**\n * Constructs an empty string builder with the specified initial [capacity].\n *\n * In Kotlin/JS
implementation of StringBuilder the initial capacity has no effect on the further performance of operations.\n */\n
 actual constructor(capacity: Int) : this() {\n }\n\n /** Constructs a string builder that contains the same
characters as the specified [content] char sequence. *\n actual constructor(content: CharSequence) :
this(content.toString()) {\n }\n\n /** Constructs an empty string builder. *\n actual constructor() :
this("")\n\n private var string: String = if (content != undefined) content else ""\n\n actual override val length: Int\n
get() = string.asDynamic().length\n\n actual override fun
get(index: Int): Char =\n string.getOrElse(index) { throw IndexOutOfBoundsException("index: $index, length:
$length") }\n\n actual override fun subSequence(startIndex: Int, endIndex: Int): CharSequence =
string.substring(startIndex, endIndex)\n\n actual override fun append(value: Char): StringBuilder {\n string +=
value\n return this\n }\n\n actual override fun append(value: CharSequence?): StringBuilder {\n string
+= value.toString()\n return this\n }\n\n actual override fun append(value: CharSequence?, startIndex: Int,
endIndex: Int): StringBuilder =\n this.appendRange(value ?: "null", startIndex, endIndex)\n\n /**\n *
Reverses the contents of this string builder and returns this instance.\n *\n * Surrogate pairs included in this
string builder are treated as single characters.\n * Therefore, the order of the high-low surrogates is never
reversed.\n *\n * Note that the reverse operation may
produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the operation.\n * For
example, reversing "\uDC00\uD800" produces "\uD800\uDC00" which is a valid surrogate pair.\n */\n
 actual fun reverse(): StringBuilder {\n var reversed = ""\n var index = string.length - 1\n while (index
>= 0) {\n val low = string[index--]\n if (low.isLowSurrogate() && index >= 0) {\n val high =
string[index--]\n if (high.isHighSurrogate()) {\n reversed = reversed + high + low\n }
else {\n reversed = reversed + low + high\n }\n } else {\n reversed += low\n }
}\n string = reversed\n return this\n }\n\n /**\n * Appends the string representation of the
specified object [value] to this string builder and returns this instance.\n *\n * The overall
effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that
string was appended to this string builder.\n *\n actual fun append(value: Any?): StringBuilder {\n string
+= value.toString()\n return this\n }\n\n /**\n * Appends the string representation of the specified boolean
[value] to this string builder and returns this instance.\n *\n * The overall effect is exactly as if the [value] were
converted to a string by the `value.toString()` method,\n * and then that string was appended to this string
builder.\n *\n @SinceKotlin("1.3")\n actual fun append(value: Boolean): StringBuilder {\n string +=
value\n return this\n }\n\n /**\n * Appends characters in the specified character array [value] to this string
builder and returns this instance.\n *\n * Characters are appended in order, starting at the index 0.\n */\n
 @SinceKotlin("1.4")\n
 @WasExperimental(ExperimentalStdlibApi::class)\n actual fun append(value: CharArray): StringBuilder {\n string +=
value.concatToString()\n return this\n }\n\n /**\n * Appends the specified string [value] to this

```

```

string builder and returns this instance.\n * \n * If [value] is `null`, then the four characters `"\null"` are
appended.\n * \n @SinceKotlin("1.3")\n actual fun append(value: String?): StringBuilder {\n this.string
+= value ?: "\null"\n return this\n }\n\n /** \n * Returns the current capacity of this string builder.\n * \n * The capacity is the maximum length this string builder can have before an allocation occurs.\n * \n * In
Kotlin/JS implementation of StringBuilder the value returned from this method may not indicate the actual size of
the backing storage.\n * \n @SinceKotlin("1.3")\n// @ExperimentalStdlibApi\n @Deprecated("Obtaining
StringBuilder capacity is
not supported in JS and common code.", level = DeprecationLevel.ERROR)\n actual fun capacity(): Int =
length\n\n /** \n * Ensures that the capacity of this string builder is at least equal to the specified
[minimumCapacity].\n * \n * If the current capacity is less than the [minimumCapacity], a new backing storage
is allocated with greater capacity.\n * Otherwise, this method takes no action and simply returns.\n * \n * In
Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given
[minimumCapacity],\n * thus calling this method has no effect on the further performance of operations.\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun
ensureCapacity(minimumCapacity: Int) {\n }\n\n /** \n * Returns the index within this string builder of the
first occurrence of the specified [string].\n * \n * Returns -1 if the specified [string] does not
occur in this string builder.\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun indexOf(string: String): Int =
this.string.asDynamic().indexOf(string)\n\n /** \n * Returns the index within this string builder of the first
occurrence of the specified [string],\n * starting at the specified [startIndex].\n * \n * Returns -1 if the
specified [string] does not occur in this string builder starting at the specified [startIndex].\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun indexOf(string: String,
startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)\n\n /** \n * Returns the index within
this string builder of the last occurrence of the specified [string].\n * The last occurrence of empty string `""` is
considered to be at the index equal to `this.length`.\n * \n * Returns -1 if the specified [string] does not occur
in
this string builder.\n * \n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n
actual fun lastIndexOf(string: String): Int = this.string.asDynamic().lastIndexOf(string)\n\n /** \n * Returns the
index within this string builder of the last occurrence of the specified [string],\n * starting from the specified
[startIndex] toward the beginning.\n * \n * Returns -1 if the specified [string] does not occur in this string
builder starting at the specified [startIndex].\n * \n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun lastIndexOf(string: String, startIndex: Int): Int {\n
if (string.isEmpty() && startIndex < 0) return -1\n return this.string.asDynamic().lastIndexOf(string,
startIndex)\n }\n\n /** \n * Inserts the string representation of the specified boolean [value] into this string
builder at the specified [index] and returns this instance.\n * \n * The
overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then
that string was inserted into this string builder at the specified [index].\n * \n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
Boolean): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n string = string.substring(0,
index) + value + string.substring(index)\n return this\n }\n\n /** \n * Inserts the specified character [value]
into this string builder at the specified [index] and returns this instance.\n * \n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n
actual fun insert(index: Int, value: Char): StringBuilder {\n AbstractList.checkPositionIndex(index,
length)\n string = string.substring(0, index) + value + string.substring(index)\n return this\n }\n\n /** \n * Inserts characters in the specified character array [value] into this string builder at the specified [index]
and returns this instance.\n * \n * The inserted characters go in same order as in the [value] character array,

```



```

string.substring(startIndex, endIndex)\n } \n\n /**\n * Attempts to reduce storage used for this string builder.\n *\n * If the backing storage of this string builder\n is larger than necessary to hold its current contents,\n * then it may be resized to become more space efficient.\n * Calling this method may, but is not required to, affect the value of the [capacity] property.\n *\n * In\n Kotlin/JS implementation of StringBuilder the size of the backing storage is always equal to the length of the string\n builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun\n trimToSize() {\n }\n\n override fun toString(): String = string\n\n /**\n * Clears the content of this string\n builder making it empty and returns this instance.\n *\n * @sample samples.text.Strings.clearStringBuilder\n *\n @SinceKotlin("1.3")\n public fun clear(): StringBuilder {\n string = ""\n return this\n }\n\n /**\n * Sets the character at the specified [index] to the specified [value].\n *\n * @throws\n IndexOutOfBoundsException if [index] is out of bounds\n of this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public operator fun set(index: Int, value: Char) {\n AbstractList.checkElementIndex(index, length)\n\n string = string.substring(0, index) + value + string.substring(index + 1)\n }\n\n /**\n * Replaces characters in\n the specified range of this string builder with characters in the specified string [value] and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to replace.\n * @param endIndex the end\n (exclusive) of the range to replace.\n * @param value the string to replace with.\n *\n * @throws\n IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length\n of this string builder, or `startIndex > endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun setRange(startIndex: Int,\n endIndex: Int, value: String): StringBuilder {\n checkReplaceRange(startIndex, endIndex, length)\n\n this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n return this\n }\n\n private fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n if (startIndex < 0 || startIndex >\n length) {\n throw IndexOutOfBoundsException("startIndex: $startIndex, length: $length")\n }\n if\n (startIndex > endIndex) {\n throw IllegalArgumentException("startIndex($startIndex) >\n endIndex($endIndex)")\n }\n }\n\n /**\n * Removes the character at the specified [index] from this string\n builder and returns this instance.\n *\n * If the `Char` at the specified [index] is part of a supplementary code\n point, this method does not remove the entire supplementary character.\n *\n * @param index the index of\n `Char` to remove.\n *\n * @throws IndexOutOfBoundsException\n if [index] is out of bounds of this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteAt(index: Int): StringBuilder {\n AbstractList.checkElementIndex(index, length)\n\n string = string.substring(0, index) + string.substring(index +\n 1)\n return this\n }\n\n /**\n * Removes characters in the specified range from this string builder and\n returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to remove.\n * @param endIndex the end (exclusive) of the range to remove.\n *\n * @throws IndexOutOfBoundsException\n or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex >\n endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun\n deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n checkReplaceRange(startIndex,\n endIndex, length)\n\n string = string.substring(0, startIndex) + string.substring(endIndex)\n return this\n }\n\n /**\n * Copies characters from this string builder into the [destination] character array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the array to copy to, 0 by\n default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n * @param\n endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n *\n * @throws\n IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this\n string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the\n subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index\n is out of the [destination] array indices range.\n
```



```

 * \n @SinceKotlin("1.4") \n @WasExperimental(ExperimentalStdlibApi::class) \n public fun
toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) { \n
 AbstractList.checkBoundsIndexes(startIndex, endIndex, length) \n
AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size) \n \n
 var dstIndex = destinationOffset \n for (index in startIndex until endIndex) { \n destination[dstIndex++]
= string[index] \n } \n } \n \n /** \n * Appends characters in a subarray of the specified character array
[value] to this string builder and returns this instance. \n * \n * Characters are appended in order, starting at
specified [startIndex]. \n * \n * @param value the array from which characters are appended. \n * @param
startIndex the beginning (inclusive) of the subarray to append. \n * @param endIndex the end (exclusive)
of the subarray to append. \n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException]
when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`. \n
* \n @SinceKotlin("1.4") \n @WasExperimental(ExperimentalStdlibApi::class) \n public fun
appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder { \n string +=
value.concatToString(startIndex, endIndex) \n return this \n } \n \n /** \n * Appends a subsequence of the
specified character sequence [value] to this string builder and returns this instance. \n * \n * @param value the
character sequence from which a subsequence is appended. \n * @param startIndex the beginning (inclusive) of
the subsequence to append. \n * @param endIndex the end (exclusive) of the subsequence to append. \n * \n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of the [value] character sequence indices or when `startIndex > endIndex`. \n * \n
@SinceKotlin("1.4") \n @WasExperimental(ExperimentalStdlibApi::class) \n public fun appendRange(value:
CharSequence, startIndex: Int, endIndex: Int): StringBuilder { \n val stringCsq = value.toString() \n
AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length) \n \n string +=
stringCsq.substring(startIndex, endIndex) \n return this \n } \n \n /** \n * Inserts characters in a subarray of
the specified character array [value] into this string builder at the specified [index] and returns this instance. \n
* \n * The inserted characters go in same order as in the [value] array, starting at [index]. \n * \n * @param index
the position in this string builder to insert at. \n * @param value the array from which characters are inserted. \n
* @param startIndex the beginning (inclusive) of the subarray to insert. \n * @param endIndex
the end (exclusive) of the subarray to insert. \n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`. \n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder. \n * \n @SinceKotlin("1.4") \n
@WasExperimental(ExperimentalStdlibApi::class) \n public fun insertRange(index: Int, value: CharArray,
startIndex: Int, endIndex: Int): StringBuilder { \n AbstractList.checkPositionIndex(index, this.length) \n \n
string = string.substring(0, index) + value.concatToString(startIndex, endIndex) + string.substring(index) \n
return this \n } \n \n /** \n * Inserts characters in a subsequence of the specified character sequence [value] into
this string builder at the specified [index] and returns this instance. \n * \n * The inserted characters go in the
same order
as in the [value] character sequence, starting at [index]. \n * \n * @param index the position in this string
builder to insert at. \n * @param value the character sequence from which a subsequence is inserted. \n *
@param startIndex the beginning (inclusive) of the subsequence to insert. \n * @param endIndex the end
(exclusive) of the subsequence to insert. \n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence
indices or when `startIndex > endIndex`. \n * @throws IndexOutOfBoundsException if [index] is less than zero or
greater than the length of this string builder. \n * \n @SinceKotlin("1.4") \n
@WasExperimental(ExperimentalStdlibApi::class) \n public fun insertRange(index: Int, value: CharSequence,
startIndex: Int, endIndex: Int): StringBuilder { \n AbstractList.checkPositionIndex(index, length) \n \n val
stringCsq = value.toString() \n

```



from which characters are appended.\n \* @param startIndex the beginning (inclusive) of the subarray to append.\n \* @param endIndex the end (exclusive) of the subarray to append.\n \* \n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER",
"NOTHING_TO_INLINE")\npublic actual inline fun StringBuilder.appendRange(value: CharArray, startIndex:
Int, endIndex: Int): StringBuilder =\n this.appendRange(value, startIndex, endIndex)\n\n/**\n * Appends a
subsequence of the specified character sequence [value] to this string builder and returns this instance.\n * \n *
@param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning
(inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out
of range of the [value] character sequence indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic
actual inline fun StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder
=\n this.appendRange(value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subarray of the specified
character array [value] into this string builder at the specified [index] and returns this instance.\n * \n * The
inserted characters go in same order as in the [value] array, starting at [index].\n * \n * @param index the
position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n *
@param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive)
of the subarray to insert.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n *
@throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string
builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n
this.insertRange(index, value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subsequence of the
specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n
*\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n
*\n * @param index the position in this string builder to insert at.\n * @param value the character sequence
from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to
insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of
the [value] character sequence indices or when `startIndex > endIndex`.\n * \n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n
this.insertRange(index, value, startIndex, endIndex)\n\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can
be found in the license/LICENSE.txt file.\n * \n * \n * Returns `true` if the content of this string is equal
to the word `true`, ignoring case, and `false` otherwise.\n * \n * @Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n * @DeprecatedSinceKotlin(hiddenSince = "1.4")\n * \n * @kotlin.internal.InlineOnly\n * \n * public actual inline fun String.toBoolean(): Boolean =
this.toBoolean()\n\n/**\n * Returns `true` if this string is not `null` and its content is equal to the word
`true`, ignoring case, and `false` otherwise.\n * \n * There are also strict versions of the function available
on non-nullable String, [toBooleanStrict] and [toBooleanStrictOrNull].\n * \n * \n * @SinceKotlin("1.4")\n * \n * public actual fun

```

String?.toBoolean(): Boolean = this != null && this.lowercase() == "true"\n\n\*\*\n \* Parses the string as a signed [Byte] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \*/\npublic actual fun String.toByte(): Byte = toByteOrNull() ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a signed [Byte] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Short] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toShort(): Short = toShortOrNull() ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Short] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toShort(radix: Int): Short = toShortOrNull(radix) ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as an [Int] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toInt(): Int = toIntOrNull() ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Long] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toInt(radix: Int): Int = toIntOrNull(radix) ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Long] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toLong(): Long = toLongOrNull() ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Long] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toLong(radix: Int): Long = toLongOrNull(radix) ?: numberFormatError(this)\n\n\*\*\n \* Parses the string as a [Double] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n \*/\npublic actual fun String.toDouble(): Double = (+this.asDynamic()).unsafeCast<Double>().also { \n if (it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())\n numberFormatError(this)\n}\n\n\*\*\n \* Parses the string as a [Float] number and returns the result.\n \* @throws NumberFormatException if the string is not a valid representation of a number.\n \* @kotlin.internal.InlineOnly\npublic actual inline fun String.toFloat(): Float = toDouble().unsafeCast<Float>()\n\n\*\*\n \* Parses the string as a [Double] number and returns the result\n \* or `null` if the string is not a valid representation of a number.\n \* @kotlin.internal.InlineOnly\npublic actual inline fun String.toDoubleOrNull(): Double? = (+this.asDynamic()).unsafeCast<Double>().takeIf { \n !(it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())\n}\n\n\*\*\n \* Parses the string as a [Float] number and returns the result\n \* or `null` if the string is not a valid representation of a number.\n \* @kotlin.internal.InlineOnly\npublic actual inline fun String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()\n\n\*\*\n \* Returns a string representation of this [Byte] value in the specified [radix].\n \* @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n \*/\n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Byte.toString(radix: Int): String = this.toInt().toString(radix)\n\n\*\*\n \* Returns a string representation of this [Short] value in the specified [radix].\n \* @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n \*/\n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Short.toString(radix: Int): String = this.toInt().toString(radix)\n\n\*\*\n \* Returns a string representation of this [Int] value in the specified [radix].\n \* @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n \*/\n@SinceKotlin("1.2")\npublic actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))\n\nprivate fun String.isNaN(): Boolean = when (this.lowercase()) {\n

```

"nan", "+nan", "-nan" -> true\n else -> false\n}\n\n/**\n * Checks whether the given [radix] is valid radix for
string to number and number to string conversion.\n */\n@PublishedApi\ninternal actual fun checkRadix(radix: Int):
Int {\n if (radix !in 2..36) {\n throw IllegalArgumentException("\radix $radix was not in valid range 2..36")\n
}\n return radix\n}\n\ninternal actual fun digitOf(char: Char, radix: Int): Int = when {\n char >= '0' && char <=
'9' -> char - '0'\n char >= 'A' && char <= 'Z' -> char - 'A' + 10\n char >= 'a' && char <= 'z' -> char
- 'a' + 10\n char < "\u0080" -> -1\n char >= "\uFF21" && char <= "\uFF3A" -> char - "\uFF21" + 10 // full-width
latin capital letter\n char >= "\uFF41" && char <= "\uFF5A" -> char - "\uFF41" + 10 // full-width latin small
letter\n else -> char.digitToIntImpl()\n}.let { if (it >= radix) -1 else it }\n","/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport
kotlin.js.RegExp\n\n/**\n * Provides enumeration values to use to set regular expression options.\n */\n\npublic actual
enum class RegexOptions(val value: String) {\n /** Enables case-insensitive matching. */\n
IGNORE_CASE("\i"),\n /** Enables multiline mode. */\n MULTILINE("\m")\n\n * In multiline mode the expressions `^` and `$`
match just after or just before, */\n * respectively, a line terminator or the end of the input sequence.\n
*\n\n private fun Iterable<RegexOption>.toFlags(prepend: String): String =
joinToString("\",", prefix = prepend) { it.value }\n\n\n /**\n * Represents the results from a single capturing group
within a [MatchResult] of [Regex].\n */\n\n * @param value The value of captured group.\n */\n\n public actual data
class MatchGroup(actual val value: String)\n\n\n /**\n * Returns a named group with the specified [name].\n */\n\n *
@return An instance of [MatchGroup] if the group with the specified [name] was matched or `null` otherwise.\n *
@throws IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.\n *
@throws UnsupportedOperationException if this match group collection doesn't support getting match groups by
name,\n * for example, when it's not supported by the current platform.\n */\n\n @SinceKotlin("1.7")\n\n public
operator fun MatchGroupCollection.get(name: String): MatchGroup? {\n val namedGroups = this as?
MatchNamedGroupCollection\n\n ?: throw UnsupportedOperationException("\Retrieving groups by name is not supported on this platform.")\n\n
return namedGroups[name]\n}\n\n\n /**\n * Represents a compiled regular expression.\n * Provides functions to
match strings in text with a pattern, replace the found occurrences and split text around matches.\n */\n\n * For pattern
syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions)\n
*\n * and [http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).\n
*\n * Note that `RegExp` objects under the hood are constructed with [the `u` flag](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/unicode)\n
*\n * that enables Unicode-related features in regular expressions. This also makes the pattern syntax more strict,\n
*\n * for example, prohibiting unnecessary escape sequences.\n */\n\n * @constructor Creates a regular expression from the specified [pattern] string and the specified
set of [options].\n */\n\n public actual class Regex actual constructor(pattern: String, options: Set<RegexOption>)\n
{\n\n /** Creates a regular expression from the specified [pattern] string and the specified single [option]. */\n
*\n\n public actual constructor(pattern: String, option: RegexOption) : this(pattern, setOf(option))\n\n\n /** Creates a
regular expression from the specified [pattern] string and the default options. */\n\n public actual
constructor(pattern: String) : this(pattern, emptySet())\n\n\n /** The pattern string of this regular expression. */\n
*\n\n public actual val pattern: String = pattern\n\n /** The set of options that were used to create this regular expression.
*/\n\n public actual val options: Set<RegexOption> = options.toSet()\n\n private val nativePattern: RegExp =
RegExp(pattern, options.toFlags("\gu"))\n\n private var nativeStickyPattern:
RegExp? = null\n\n private fun initStickyPattern(): RegExp =\n\n nativeStickyPattern ?: RegExp(pattern,
options.toFlags("\yu")).also { nativeStickyPattern = it }\n\n\n private var nativeMatchesEntirePattern: RegExp? =
null\n\n private fun initMatchesEntirePattern(): RegExp =\n\n nativeMatchesEntirePattern ?: run {\n\n if
(pattern.startsWith('^') && pattern.endsWith('$'))\n\n nativePattern\n\n else\n\n return

```

```

RegExp(\^{\$pattern.trimStart('^').trimEnd('$')}\$", options.toFlags("gu"))\n }.also {
nativeMatchesEntirePattern = it }\n\n\n /** Indicates whether the regular expression matches the entire [input].
*\n public actual infix fun matches(input: CharSequence): Boolean {\n nativePattern.reset()\n val match
= nativePattern.exec(input.toString())\n return match != null && match.index == 0 && nativePattern.lastIndex
== input.length\n }\n\n /** Indicates whether the regular expression
can find at least one match in the specified [input]. *\n public actual fun containsMatchIn(input: CharSequence):
Boolean {\n nativePattern.reset()\n return nativePattern.test(input.toString())\n }\n\n
@SinceKotlin("1.7")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual fun
matchesAt(input: CharSequence, index: Int): Boolean {\n if (index < 0 || index > input.length) {\n throw
IndexOutOfBoundsException("index out of bounds: \$index, input length: \${input.length}")\n }\n val
pattern = initStickyPattern()\n pattern.lastIndex = index\n return pattern.test(input.toString())\n }\n\n
/**\n * Returns the first match of a regular expression in the [input], beginning at the specified [startIndex].\n
*\n * @param startIndex An index to start search with, by default 0. Must be not less than zero and not greater
than `input.length`\n * @return An instance of [MatchResult] if match
was found or `null` otherwise.\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or
greater than the length of the [input] char sequence.\n * @sample samples.text.Regexps.find\n */\n
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun find(input:
CharSequence, startIndex: Int = 0): MatchResult? {\n if (startIndex < 0 || startIndex > input.length) {\n
throw IndexOutOfBoundsException("Start index out of bounds: \$startIndex, input length: \${input.length}")\n
}\n return nativePattern.findNext(input.toString(), startIndex, nativePattern)\n }\n\n /**\n * Returns a
sequence of all occurrences of a regular expression within the [input] string, beginning at the specified
[startIndex].\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the
length of the [input] char sequence.\n *\n * @sample samples.text.Regexps.findAll\n */\n
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun findAll(input:
CharSequence, startIndex: Int = 0): Sequence<MatchResult> {\n if (startIndex < 0 || startIndex > input.length)
{\n throw IndexOutOfBoundsException("Start index out of bounds: \$startIndex, input length:
\${input.length}")\n }\n return generateSequence({ find(input, startIndex) }, { match -> match.next() })\n
}\n\n /**\n * Attempts to match the entire [input] CharSequence against the pattern.\n *\n * @return An
instance of [MatchResult] if the entire input matches or `null` otherwise.\n */\n public actual fun
matchEntire(input: CharSequence): MatchResult? =\n initMatchesEntirePattern().findNext(input.toString(), 0,
nativePattern)\n\n @SinceKotlin("1.7")\n @WasExperimental(ExperimentalStdlibApi::class)\n public actual
fun matchAt(input: CharSequence, index: Int): MatchResult? {\n if (index < 0 || index > input.length)
{\n throw IndexOutOfBoundsException("index out of bounds: \$index, input length: \${input.length}")\n
}\n return initStickyPattern().findNext(input.toString(), index, nativePattern)\n }\n\n\n /**\n * Replaces
all occurrences of this regular expression in the specified [input] string with specified [replacement] expression.\n
*\n * The replacement string may contain references to the captured groups during a match. Occurrences of
`{\$name}` or `\$index`\n * in the replacement string will be substituted with the subsequences corresponding to
the captured groups with the specified name or index.\n * In case of `\$index`, the first digit after '$' is always
treated as a part of group reference. Subsequent digits are incorporated\n * into `index` only if they would form a
valid group reference. Only the digits '0'..'9' are considered as potential components\n * of the group reference.
Note that indexes of captured groups start from
1, and the group with index 0 is the whole match.\n * In case of `{\$name}`, the `name` can consist of latin letters
'a'..'z' and 'A'..'Z', or digits '0'..'9'. The first character must be\n * a letter.\n *\n * Backslash character '\\' can
be used to include the succeeding character as a literal in the replacement string, e.g. '\\\$` or '\\\\\\'.\n *
[Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n *\n * @param
input the char sequence to find matches of this regular expression in\n * @param replacement the expression to
replace found matches with\n * @return the result of replacing each occurrence of this regular expression in
[input] with the result of evaluating the [replacement] expression\n * @throws RuntimeException if

```

[replacement] expression is malformed, or capturing group with specified `name` or `index` does not exist

```

public actual fun replace(input: CharSequence, replacement: String):
String {
 if (!replacement.contains("\\\$") && !replacement.contains('$')) {
 return
input.toString().nativeReplace(nativePattern, replacement)
 }
 return replace(input) {
substituteGroupRefs(it, replacement)
}
}
/**
 * Replaces all occurrences of this regular expression in
the specified [input] string with the result of
 * the given function [transform] that takes [MatchResult] and
returns a string to be used as a
 * replacement for that match.
 */
public actual fun replace(input:
CharSequence, transform: (MatchResult) -> CharSequence): String {
 var match = find(input)
 if (match
== null) return input.toString()
 var lastStart = 0
 val length = input.length
 val sb =
StringBuilder(length)
 do {
 val foundMatch = match!!
 sb.append(input, lastStart,
foundMatch.range.start)
 sb.append(transform(foundMatch))
 lastStart
= foundMatch.range.endInclusive + 1
 match = foundMatch.next()
 } while (lastStart < length &&
match != null)
 if (lastStart < length) {
 sb.append(input, lastStart, length)
 }
 return
sb.toString()
}
/**
 * Replaces the first occurrence of this regular expression in the specified [input]
string with specified [replacement] expression.
 * The replacement string may contain references to the
captured groups during a match. Occurrences of `${name}` or `${index}`
 * in the replacement string will be
substituted with the subsequences corresponding to the captured groups with the specified name or index.
 * In
case of `${index}`, the first digit after '$' is always treated as a part of group reference. Subsequent digits are
incorporated
 * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered
as potential components
 * of the group
reference. Note that indexes of captured groups start from 1, and the group with index 0 is the whole match.
 * In
case of `${name}`, the `name` can consist of latin letters 'a'..'z' and 'A'..'Z', or digits '0'..'9'. The first character
must be
 * a letter.
 * Backslash character '\' can be used to include the succeeding character as a
literal in the replacement string, e.g. `\$` or `\\`.
 * [Regex.escapeReplacement] can be used if [replacement]
have to be treated as a literal string.
 * @param input the char sequence to find a match of this regular
expression in
 * @param replacement the expression to replace the found match with
 * @return the result of
replacing the first occurrence of this regular expression in [input] with the result of evaluating the [replacement]
expression
 * @throws RuntimeException if [replacement] expression is malformed, or capturing group with
specified `name` or `index` does not exist
 */
public actual fun replaceFirst(input: CharSequence, replacement: String): String {
 if
(!replacement.contains("\\\$") && !replacement.contains('$')) {
 val nonGlobalOptions =
options.toFlags("u")
 return input.toString().nativeReplace(RegExp(pattern, nonGlobalOptions),
replacement)
 }
 val match = find(input) ?: return input.toString()
 return buildString {
append(input.substring(0, match.range.first))
 append(substituteGroupRefs(match, replacement))
append(input.substring(match.range.last + 1, input.length))
 }
}
/**
 * Splits the [input]
CharSequence to a list of strings around matches of this regular expression.
 * @param limit Non-negative
value specifying the maximum number of substrings the string can be split to.
 * Zero by default means no limit
is set.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual
fun split(input: CharSequence, limit: Int = 0): List<String> {
 requireNonNegativeLimit(limit)
 val
matches = findAll(input).let { if (limit == 0) it else it.take(limit - 1) }
 val result = mutableListOf<String>()
 var lastStart = 0
 for (match in matches) {
 result.add(input.subSequence(lastStart,
match.range.start).toString())
 lastStart = match.range.endInclusive + 1
 }
 result.add(input.subSequence(lastStart, input.length).toString())
 return result
}
/**
 * Splits the
[input] CharSequence to a sequence of strings around matches of this regular expression.
 * @param limit
Non-negative value specifying the maximum number of substrings the string can be split to.
 * Zero by default
means no limit is set.
 * @sample samples.text.Regexps.splitToSequence
 */
@SinceKotlin("1.6")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")

```

```

public actual fun splitToSequence(input: CharSequence, limit: Int = 0): Sequence<String> {
 requireNonNegativeLimit(limit)
 return sequence {
 var match = find(input)
 if (match == null || limit == 1) {
 yield(input.toString())
 return@sequence
 }
 var nextStart = 0
 var splitCount = 0
 do {
 val foundMatch = match!!
 yield(input.substring(nextStart, foundMatch.range.first))
 nextStart = foundMatch.range.endInclusive + 1
 match = foundMatch.next()
 } while (++splitCount != limit - 1 && match != null)
 yield(input.substring(nextStart, input.length))
 }
}

/**
 * Returns the string representation of this regular expression, namely the [pattern] of this regular expression.
 * Note that another regular expression constructed from the same pattern string may have different [options] and may match strings differently.
 */
public override fun toString(): String = nativePattern.toString()

actual companion object {
 /**
 * Returns a regular expression that matches the specified [literal] string literally.
 * No characters of that string will have special meaning when searching for an occurrence of the regular expression.
 */
 public actual fun fromLiteral(literal: String): Regex = Regex(escape(literal))

 /**
 * Returns a regular expression pattern string that matches the specified [literal] string literally.
 * No characters of that string will have special meaning when searching for an occurrence of the regular expression.
 */
 public actual fun escape(literal: String): String = literal.nativeReplace(patternEscape, "\\|\\$&\\|")

 /**
 * Returns a literal replacement expression for the specified [literal] string.
 * No characters of that string will have special meaning when it is used as a replacement string in [Regex.replace] function.
 */
 public actual fun escapeReplacement(literal: String): String = literal.nativeReplace(replacementEscape, "\\|\\$&\\|")

 private val patternEscape = Regex("\\|\\$*+?.()\\[\\]{}\\|\\|", "g")
 private val replacementEscape = Regex("\\|\\$\\|\\|", "g")

 internal fun nativeEscapeReplacement(literal: String): String = literal.nativeReplace(nativeReplacementEscape, "\\$\\$\\$")

 private val nativeReplacementEscape = Regex("\\|\\$\\|\\|", "g")

 private fun Regex.findNext(input: String, from: Int, nextPattern: Regex): MatchResult? {
 this.lastIndex = from
 val match = exec(input)
 if (match == null) return null
 val range = match.index..lastIndex - 1
 return object : MatchResult {
 override val range: IntRange = range
 override val value: String = match[0]!!
 override val groups: MatchGroupCollection = object : MatchNamedGroupCollection, AbstractCollection<MatchGroup?>() {
 override val size: Int get() = match.length
 override fun iterator(): Iterator<MatchGroup?> = indices.asSequence().map { this[it] }.iterator()
 override fun get(index: Int): MatchGroup? = match[index]
 let { MatchGroup(it) }
 override fun get(name: String): MatchGroup? {
 // An object of named capturing groups whose keys are the names and values are the capturing groups
 // or undefined if no named capturing groups were defined.
 val groups = match.asDynamic().groups
 ?: throw IllegalArgumentException("Capturing group with name { $name } does not exist. No named capturing group was defined in Regex")
 // If the match was successful but the group specified failed to match any part of the input sequence, the associated value is 'undefined'. Value for a non-existent key is also 'undefined'. Thus, explicitly check if the key exists.
 if (!hasOwnProperty(groups, name)) throw IllegalArgumentException("Capturing group with name { $name } does not exist")
 val value = groups[name]
 return if (value == undefined) null else MatchGroup(value as String)
 }
 }
 private fun hasOwnProperty(o: Any?, name: String): Boolean {
 return js("Object").prototype.hasOwnProperty.call(o, name).unsafeCast<Boolean>()
 }
 private var groupValues_: List<String>? = null
 override val groupValues: List<String> get() {
 if (groupValues_ == null) {
 groupValues_ = object : AbstractList<String>() {
 override val size: Int get() = match.length
 override fun get(index: Int): String = match[index]
 }
 }
 return groupValues_!!
 }
 override fun next(): MatchResult? = nextPattern.findNext(input, if (range.isEmpty()) advanceToNextCharacter(range.start) else range.endInclusive + 1, nextPattern)
 private fun

```



```

advanceToNextCharacter(index: Int): Int {
 if (index < input.lastIndex) {
 val code1 = input.asDynamic().charCodeAt(index).unsafeCast<Int>()
 if (code1 in 0xD800..0xDBFF) {
 val code2 = input.asDynamic().charCodeAt(index + 1).unsafeCast<Int>()
 if (code2 in 0xDC00..0xDFFF) {
 return index + 2
 }
 }
 return index + 1
 }
}

// The same code from K/N Regex.kt
private fun substituteGroupRefs(match: MatchResult, replacement: String): String {
 var index = 0
 val result = StringBuilder()
 while (index < replacement.length) {
 val char = replacement[index++]
 if (char == '\\') {
 if (index == replacement.length)
 throw IllegalArgumentException("The Char to be escaped is missing")
 result.append(replacement[index++])
 } else if (char == '$') {
 if (index == replacement.length)
 throw IllegalArgumentException("Capturing group index is missing")
 if (replacement[index] == '{') {
 val endIndex = replacement.readGroupName(++index)
 if (index == endIndex)
 throw IllegalArgumentException("Named capturing group reference should have a non-empty name")
 if (endIndex == replacement.length || replacement[endIndex] != '}')
 throw IllegalArgumentException("Named capturing group reference is missing trailing '}'")
 val groupName = replacement.substring(index, endIndex)
 result.append(match.groups[groupName]?.value ?: "\\")
 index = endIndex + 1 // skip past '}'
 } else {
 if (replacement[index] !in '0'..'9')
 throw IllegalArgumentException("Invalid capturing group reference")
 val groups = match.groups
 val endIndex = replacement.readGroupIndex(index, groups.size)
 val groupIndex = replacement.substring(index, endIndex).toInt()
 if (groupIndex >= groups.size)
 throw IndexOutOfBoundsException("Group with index $groupIndex does not exist")
 result.append(groups[groupIndex]?.value ?: "\\")
 index = endIndex
 }
 } else {
 result.append(char)
 }
 }
 return result.toString()
}

// The name must be a legal JavaScript identifier. See https://262.ecma-international.org/5.1/#sec-7.6
// Don't try to validate the referenced group name as it may be time-consuming
// If the name is invalid, it won't be found in `match.groups` anyway and will throw
// Group names in the target Regex are validated at creation time
private fun String.readGroupName(startIndex: Int): Int {
 var index = startIndex
 while (index < length) {
 if (this[index] == '}')
 break
 index++
 }
 return index
}

private fun String.readGroupIndex(startIndex: Int, groupCount: Int): Int {
 // at least one digit after '$' is always captured
 var index = startIndex + 1
 var groupIndex = this[startIndex] - '0'
 // capture the largest valid group index
 while (index < length && this[index] in '0'..'9') {
 val newGroupIndex = (groupIndex * 10) + (this[index] - '0')
 if (newGroupIndex in 0 until groupCount) {
 groupIndex = newGroupIndex
 index++
 }
 }
 return index
}

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")
@file:Suppress("EXTENSION_SHADOWED_BY_MEMBER")
package kotlin.text
import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
 /** Constructs an empty string builder. */
 constructor()

 /** Constructs an empty string builder with the specified initial [capacity]. */
 constructor(capacity: Int)

 /** Constructs a string builder that contains the same characters as the specified [content] char sequence. */
 constructor(content: CharSequence)

 /** Constructs a string builder that contains the same characters as the specified [content] string. */
 @SinceKotlin("1.3")
 @ExperimentalStdlibApi
 constructor(content: String)

 override val length: Int

 override operator fun get(index: Int): Char

 override fun subSequence(startIndex: Int, endIndex: Int): CharSequence

 override fun append(value: Char): StringBuilder

```

```

override fun append(value: CharSequence?): StringBuilder\n override fun append(value: CharSequence?,
startIndex: Int, endIndex: Int): StringBuilder\n\n /**\n * Reverses the contents of this string builder and returns
this instance.\n *\n * Surrogate pairs included in this string builder are treated as single characters.\n *\n * Therefore, the order of the high-low surrogates is never reversed.\n *\n * Note that
the reverse operation may produce new surrogate pairs that were unpaired low-surrogates and high-surrogates
before the operation.\n * For example, reversing `"\uDC00\uD800"` produces `"\uD800\uDC00"` which is a
valid surrogate pair.\n */\n fun reverse(): StringBuilder\n\n /**\n * Appends the string representation of the
specified object [value] to this string builder and returns this instance.\n *\n * The overall effect is exactly as if
the [value] were converted to a string by the `value.toString()` method,\n * and then that string was appended to
this string builder.\n */\n fun append(value: Any?): StringBuilder\n\n /**\n * Appends the string
representation of the specified boolean [value] to this string builder and returns this instance.\n *\n * The
overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then
that string was appended to this string builder.\n */\n @SinceKotlin("1.3")\n fun append(value: Boolean): StringBuilder\n\n /**\n * Appends characters in
the specified character array [value] to this string builder and returns this instance.\n *\n * Characters are
appended in order, starting at the index 0.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun append(value: CharArray): StringBuilder\n\n /**\n *
Appends the specified string [value] to this string builder and returns this instance.\n *\n * If [value] is `null`,
then the four characters `"\u0000\u0000\u0000\u0000"` are appended.\n */\n @SinceKotlin("1.3")\n fun append(value: String?):
StringBuilder\n\n /**\n * Returns the current capacity of this string builder.\n *\n * The capacity is the
maximum length this string builder can have before an allocation occurs.\n */\n @SinceKotlin("1.3")\n @ExperimentalStdlibApi\n @Deprecated("Obtaining StringBuilder capacity is not
supported in JS and common code.", level = DeprecationLevel.ERROR)\n fun capacity(): Int\n\n /**\n *
Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].\n *\n * If
the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n *
Otherwise, this method takes no action and simply returns.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun ensureCapacity(minimumCapacity: Int)\n\n /**\n *
Returns the index within this string builder of the first occurrence of the specified [string].\n *\n * Returns `-1`
if the specified [string] does not occur in this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun indexOf(string: String): Int\n\n /**\n * Returns the
index within this string builder of the first occurrence of the specified [string],\n *\n * starting at the specified [startIndex].\n *\n * Returns `-1` if the specified [string] does not occur in this
string builder starting at the specified [startIndex].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun indexOf(string: String, startIndex: Int): Int\n\n /**\n *
Returns the index within this string builder of the last occurrence of the specified [string].\n * The last
occurrence of empty string `""` is considered to be at the index equal to `this.length`.\n *\n * Returns `-1` if
the specified [string] does not occur in this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun lastIndexOf(string: String): Int\n\n /**\n * Returns
the index within this string builder of the last occurrence of the specified [string],\n * starting from the specified
[startIndex] toward the beginning.\n *\n * Returns `-1` if the specified
[string] does not occur in this string builder starting at the specified [startIndex].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun lastIndexOf(string: String, startIndex: Int): Int\n\n /**\n *
Inserts the string representation of the specified boolean [value] into this string builder at the specified
[index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string
by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified
[index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of
this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun
insert(index: Int, value: Boolean): StringBuilder\n\n /**\n * Inserts the specified character [value] into this

```

```

string builder at the specified [index]
and returns this instance.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater
than the length of this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Char): StringBuilder\n\n /**\n * Inserts characters in the specified character array [value] into this string builder at the specified [index] and
returns this instance.\n *\n * The inserted characters go in same order as in the [value] character array, starting
at [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length
of this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n
fun insert(index: Int, value: CharArray): StringBuilder\n\n /**\n * Inserts characters in the specified character
sequence [value] into this string builder at the specified [index] and returns this
instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence, starting at
[index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character
sequence from which characters are inserted. If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n
*\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string
builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun
insert(index: Int, value: CharSequence?): StringBuilder\n\n /**\n * Inserts the string representation of the
specified object [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then
that string was inserted into this string builder
at the specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater
than the length of this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Any?): StringBuilder\n\n /**\n *
Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n *\n * If
[value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n *\n * @throws IndexOutOfBoundsException
if [index] is less than zero or greater than the length of this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: String?): StringBuilder\n\n
/**\n * Sets the length of this string builder to the specified [newLength].\n *\n * If the [newLength] is less
than the current length, it is changed to the specified [newLength].\n * Otherwise,
null characters '\u0000' are appended to this string builder until its length is less than the [newLength].\n *\n *
Note that in Kotlin/JS [set] operator function has non-constant execution time complexity.\n * Therefore,
increasing length of this string builder and then updating each character by index may slow down your program.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n
*\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun setLength(newLength:
Int)\n\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive)
and up to the [length] (exclusive).\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than
zero or greater than the length of this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun substring(startIndex:
Int): String\n\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex]
(inclusive) and up to the [endIndex] (exclusive).\n *\n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when
`startIndex > endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n
fun substring(startIndex: Int, endIndex: Int): String\n\n /**\n * Attempts to reduce storage used for this string
builder.\n *\n * If the backing storage of this string builder is larger than necessary to hold its current
contents,\n * then it may be resized to become more space efficient.\n * Calling this method may, but is not
required to, affect the value of the [capacity] property.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun trimToSize()\n\n /**\n * Clears
the content of this string builder making it empty and returns this instance.\n *\n * @sample
samples.text.Strings.clearStringBuilder\n *\n @SinceKotlin("1.3")\n\n public expect fun StringBuilder.clear():

```

`StringBuilder`\n\n/\*\*\n \* Sets the character at the specified [index] to the specified [value].\n \*\n \* @throws  
 IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect operator fun  
`StringBuilder.set(index: Int, value: Char)`\n\n/\*\*\n \* Replaces characters in the specified range of this string builder  
 with characters in the specified string [value] and returns this instance.\n \*\n \* @param startIndex the beginning  
 (inclusive) of the range to replace.\n \* @param endIndex the end (exclusive) of the range to replace.\n \* @param  
 value the string to replace with.\n \*\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] if  
 [startIndex] is less than zero, greater than  
 the length of this string builder, or `startIndex > endIndex`.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder`\n\n/\*\*\n \* Removes the  
 character at the specified [index] from this string builder and returns this instance.\n \*\n \* If the `Char` at the  
 specified [index] is part of a supplementary code point, this method does not remove the entire supplementary  
 character.\n \*\n \* @param index the index of `Char` to remove.\n \*\n \* @throws IndexOutOfBoundsException if  
 [index] is out of bounds of this string builder.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.deleteAt(index: Int): StringBuilder`\n\n/\*\*\n \* Removes characters in the specified range from this  
 string builder and returns this instance.\n \*\n \* @param startIndex the beginning (inclusive) of the range to  
 remove.\n \* @param endIndex the  
 end (exclusive) of the range to remove.\n \*\n \* @throws IndexOutOfBoundsException or  
 [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex >  
 endIndex`.\n \*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder`\n\n/\*\*\n \* Copies characters from this  
 string builder into the [destination] character array.\n \*\n \* @param destination the array to copy to.\n \* @param  
 destinationOffset the position in the array to copy to, 0 by default.\n \* @param startIndex the beginning (inclusive)  
 of the range to copy, 0 by default.\n \* @param endIndex the end (exclusive) of the range to copy, length of this  
 string builder by default.\n \*\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when  
 [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n \*\n \* @throws  
 IndexOutOfBoundsException  
 when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n \*\n \* or when  
 that index is out of the [destination] array indices range.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =  
 this.length)`\n\n/\*\*\n \* Appends characters in a subarray of the specified character array [value] to this string builder  
 and returns this instance.\n \*\n \* Characters are appended in order, starting at specified [startIndex].\n \*\n \* @param  
 value the array from which characters are appended.\n \* @param startIndex the beginning (inclusive) of the  
 subarray to append.\n \* @param endIndex the end (exclusive) of the subarray to append.\n \*\n \* @throws  
 IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the  
 [value] array indices or when `startIndex  
 > endIndex`.\n \*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder`\n\n/\*\*\n \* Appends a  
 subsequence of the specified character sequence [value] to this string builder and returns this instance.\n \*\n \*\n  
 @param value the character sequence from which a subsequence is appended.\n \* @param startIndex the beginning  
 (inclusive) of the subsequence to append.\n \* @param endIndex the end (exclusive) of the subsequence to append.\n

\*\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out  
 of range of the [value] character sequence indices or when `startIndex > endIndex`.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun  
`StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder`\n\n/\*\*\n \* Inserts

characters in a subarray of the specified

character array [value] into this string builder at the specified [index] and returns this instance.\n \*\n \* The inserted characters go in same order as in the [value] array, starting at [index].\n \*\n \* @param index the position in this string builder to insert at.\n \* @param value the array from which characters are inserted.\n \* @param startIndex the beginning (inclusive) of the subarray to insert.\n \* @param endIndex the end (exclusive) of the subarray to insert.\n \*\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

```
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n Inserts characters
```

in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n \*\n \* The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n \*\n \* @param index the position in this string builder to insert at.\n \* @param value the character sequence from which a subsequence is inserted.\n \* @param startIndex the beginning (inclusive) of the subsequence to insert.\n \* @param endIndex the end (exclusive) of the subsequence to insert.\n \*\n \* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n \* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

```
StringBuilder.insertRange(index: Int,
value: CharSequence, startIndex: Int, endIndex: Int):
```

```
StringBuilder\n\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\n@Deprecated("Use
append(value: Any?) instead", ReplaceWith("append(value = obj)"),
```

```
DeprecationLevel.WARNING)\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.append(obj: Any?):
StringBuilder = this.append(obj)\n\n/**\n Builds new string by populating newly created [StringBuilder] using
provided [builderAction]\n * and then converting it to [String].\n *\n@kotlin.internal.InlineOnly\npublic inline fun
buildString(builderAction: String Builder.() -> Unit): String {\n contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }\n return StringBuilder().apply(builderAction).toString()\n}\n\n/**\n Builds new string by populating newly created [StringBuilder] initialized with the given [capacity]\n * using
provided [builderAction] and then converting it to [String].\n
```

```
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun buildString(capacity:
Int, builderAction: String Builder.() -> Unit): String {\n contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }\n return
```

```
StringBuilder(capacity).apply(builderAction).toString()\n}\n\n/**\n Appends all arguments to the given
StringBuilder.\n *\n@public fun StringBuilder.append(vararg value: String?): String Builder {\n for (item in
value)\n append(item)\n return this\n}\n\n/**\n Appends all arguments to the given String Builder.\n
```

```
*\n@public fun StringBuilder.append(vararg value: Any?): String Builder {\n for (item in value)\n
```

```
append(item)\n return this\n}\n\n// KT-52336\n@Deprecated("Use appendRange instead.",
```

```
ReplaceWith("this.appendRange(str, offset, offset + len)"), level =
```

```
DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\n@Suppress("UNUSED_PARAMETER")\npublic inline
fun StringBuilder.append(str: CharArray, offset: Int, len: Int): String Builder = throw NotImplementedError()\n\n/**
```

```
Appends a line feed character
```

```
(`\\n`) to this String Builder. *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
StringBuilder.appendLine(): String Builder = append("`\\n`)\n\n/**\n Appends [value] to this [StringBuilder], followed
by a line feed character (`\\n`). *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
StringBuilder.appendLine(value: CharSequence?): String Builder = append(value).appendLine()\n\n/**\n Appends
```

```
[value] to this [StringBuilder], followed by a line feed character (`\\n`).
```

```

*^@SinceKotlin("1.4")^@kotlin.internal.InlineOnly^public inline fun StringBuilder.appendLine(value:
String?): StringBuilder = append(value).appendLine()\n/** Appends [value] to this [StringBuilder], followed by a
line feed character (`\n`). */^@SinceKotlin("1.4")^@kotlin.internal.InlineOnly^public inline fun
StringBuilder.appendLine(value: Any?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this
[StringBuilder], followed by a line feed character
(`\n`). */^@SinceKotlin("1.4")^@kotlin.internal.InlineOnly^public inline fun StringBuilder.appendLine(value:
CharArray): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed
by a line feed character (`\n`). */^@SinceKotlin("1.4")^@kotlin.internal.InlineOnly^public inline fun
StringBuilder.appendLine(value: Char): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this
[StringBuilder], followed by a line feed character (`\n`).
*/^@SinceKotlin("1.4")^@kotlin.internal.InlineOnly^public inline fun StringBuilder.appendLine(value:
Boolean): StringBuilder = append(value).appendLine()\n", /*^n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */^@npackage kotlin.text\n\nimport
kotlin.js.RegExp\n\n@kotlin.internal.InlineOnly^internal actual inline fun String.nativeIndexOf(ch:
Char, fromIndex: Int): Int = nativeIndexOf(ch.toString(), fromIndex)\n\n@kotlin.internal.InlineOnly^internal
actual inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int = nativeLastIndexOf(ch.toString(),
fromIndex)\n\n/**\n * Returns `true` if this string starts with the specified prefix.\n
*/^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^public actual fun
String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return
nativeStartsWith(prefix, 0)\n else\n return regionMatches(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.\n
*/^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^public actual fun
String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return nativeStartsWith(prefix, startIndex)\n
 else\n return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if
this string ends with the specified suffix.\n
*/^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^public actual fun
String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return
nativeEndsWith(suffix)\n else\n return regionMatches(length - suffix.length, suffix, 0, suffix.length,
ignoreCase)\n}\n\n@Deprecated("Use Regex.matches() instead",
ReplaceWith("regex.toRegex().matches(this)"))^@DeprecatedSinceKotlin(warningSince = "1.6")^public fun
String.matches(regex: String): Boolean {\n @Suppress("DEPRECATION")\n val result = this.match(regex)\n return result != null && result.size != 0\n}\n\n/**\n * Returns `true` if this string is empty or consists solely of
whitespace characters.\n
*/\n * @sample samples.text.Strings.stringIsBlank\n */^public actual fun
CharSequence.isBlank():
Boolean = length == 0 || indices.all { this[it].isWhitespace() }\n\n/**\n * Returns `true` if this string is equal to
[other], optionally ignoring character case.\n
*/\n * Two strings are considered to be equal if they have the same
length and the same character at the same index.\n
*/\n * If [ignoreCase] is true, the result of
`Char.toUpperCaseChar().toLowerCaseChar()` on each character is compared.\n
*/\n * @param ignoreCase `true` to ignore
character case when comparing strings. By default `false`.\n
*/^@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^public actual fun
String?.equals(other: String?, ignoreCase: Boolean = false): Boolean {\n if (this == null) return other == null\n if
(other == null) return false\n if (!ignoreCase) return this == other\n if (this.length != other.length) return
false\n for (index in 0 until this.length) {\n val thisChar = this[index]\n val otherChar = other[index]\n if (!thisChar.equals(otherChar, ignoreCase)) {\n
 return false\n }\n }\n return
true\n}\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")^public actual fun

```

```

CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase:
Boolean = false): Boolean =\n regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)\n\n/**\n *
Returns a copy of this string having its first letter titlecased using the rules of the default locale,\n * or the original
string if it's empty or already starts with a title case letter.\n * The title case of a character is usually the same as
its upper case with several exceptions.\n * The particular list of characters with the special title case form depends
on the underlying platform.\n * @sample samples.text.Strings.capitalize\n */\n@Deprecated("Use
replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase()) it.titlecase() else it.toString()
}"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\npublic actual fun String.capitalize(): String {\n return if (isEmpty()) substring(0, 1).uppercase() +
substring(1) else this\n}\n\n/**\n * Returns a copy of this string having its first letter lowercased using the rules of
the default locale,\n * or the original string if it's empty or already starts with a lower case letter.\n * @sample
samples.text.Strings.decapitalize\n */\n@Deprecated("Use replaceFirstChar instead.",
ReplaceWith("replaceFirstChar { it.lowercase() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
actual fun String.decapitalize(): String {\n return if (isEmpty()) substring(0, 1).lowercase() + substring(1) else
this\n}\n\n/**\n * Returns a string containing this char sequence repeated [n] times.\n * @throws
[IllegalArgumentException] when n < 0.\n * @sample samples.text.Strings.repeat\n */\npublic actual fun
CharSequence.repeat(n: Int): String {\n require(n >= 0) { "Count 'n' must be non-negative, but was $n." }\n
return
when (n) {\n 0 -> ""\n 1 -> this.toString()\n else -> {\n var result = ""\n if (!isEmpty())
{\n var s = this.toString()\n var count = n\n while (true) {\n if ((count and 1)
== 1) {\n result += s\n }\n count = count ushr 1\n if (count == 0)
{\n break\n }\n s += s\n }\n return result\n }\n }\n}\n\n/**\n * Returns a new string obtained by replacing all occurrences of the [oldValue] substring in this
string\n * with the specified [newValue] string.\n * @sample samples.text.Strings.replace\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n nativeReplace(Regex.escape(oldValue),
if (ignoreCase) "gui" else "gu"), Regex.nativeEscapeReplacement(newValue))\n\n/**\n * Returns a new string
with all occurrences of [oldChar] replaced with [newChar].\n * @sample samples.text.Strings.replace\n */\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n nativeReplace(Regex.escape(oldChar.toString()), if (ignoreCase) "gui" else "gu"),
newChar.toString())\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual
fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n nativeReplace(Regex.escape(oldValue), if (ignoreCase) "ui" else "u"),
Regex.nativeEscapeReplacement(newValue))\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGU
MENTS")\npublic actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false):
String =\n nativeReplace(Regex.escape(oldChar.toString()),
if (ignoreCase) "ui" else "u"), newChar.toString())\n\n/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/** Returns the negative [size] if
[throwOnMalformed] is false, throws [CharacterCodingException] otherwise. */\nprivate fun malformed(size: Int,
index: Int, throwOnMalformed: Boolean): Int {\n if (throwOnMalformed) throw
CharacterCodingException("Malformed sequence starting at ${index - 1}")\n return -size\n}\n\n/** Returns
code point corresponding to UTF-16 surrogate pair,\n * where the first of the pair is the [high] and the second is in
the [string] at the [index].\n * Returns zero if the pair is malformed and [throwOnMalformed] is false.\n * @throws
CharacterCodingException if the pair is malformed and [throwOnMalformed] is true.\n */\nprivate

```

```

fun codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int
{
 if (high !in 0xD800..0xDBFF || index >= endIndex) {
 return malformed(0, index, throwOnMalformed)
 }
 val low = string[index].code
 if (low !in 0xDC00..0xDFFF) {
 return malformed(0, index, throwOnMalformed)
 }
 return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)
}

Returns code point corresponding to UTF-8 sequence of two bytes, where the first byte of the sequence is the
[byte1] and the second byte is in the [bytes] array at the [index]. Returns zero if the sequence is malformed and
[throwOnMalformed] is false. @throws CharacterCodingException if the sequence of two bytes is
malformed and [throwOnMalformed] is true.

private fun codePointFrom2(bytes: ByteArray, byte1: Int, index:
Int, endIndex: Int, throwOnMalformed: Boolean): Int {
 if (byte1 and 0x1E == 0 || index >= endIndex)
 {
 return malformed(0, index, throwOnMalformed)
 }
 val byte2 = bytes[index].toInt()
 if (byte2 and 0xC0 != 0x80)
 {
 return malformed(0, index, throwOnMalformed)
 }
 return (byte1 shl 6) xor byte2 xor 0xF8
}

Returns code point corresponding to UTF-8 sequence of three bytes, where the first byte of
the sequence is the [byte1] and the others are in the [bytes] array starting from the [index]. Returns a non-
positive value indicating number of bytes from [bytes] included in malformed sequence if the sequence is
malformed and [throwOnMalformed] is false. @throws CharacterCodingException if the sequence of three
bytes is malformed and [throwOnMalformed] is true.

private fun codePointFrom3(bytes: ByteArray, byte1:
Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
 if (index >= endIndex) {
 return
malformed(0, index, throwOnMalformed)
 }
 val byte2 = bytes[index].toInt()
 if (byte1
and 0xF == 0) {
 if (byte2 and 0xE0 != 0xA0) {
 // Non-shortest form
 return malformed(0,
index, throwOnMalformed)
 }
 } else if (byte1 and 0xF == 0xD) {
 if (byte2 and 0xE0 != 0x80) {
 // Surrogate code point
 return malformed(0, index, throwOnMalformed)
 }
 } else if (byte2 and
0xC0 != 0x80) {
 return malformed(0, index, throwOnMalformed)
 }
 if (index + 1 == endIndex) {
 return malformed(1, index, throwOnMalformed)
 }
 val byte3 = bytes[index + 1].toInt()
 if (byte3 and
0xC0 != 0x80) {
 return malformed(1, index, throwOnMalformed)
 }
 return (byte1 shl 12) xor (byte2
shl 6) xor byte3 xor -0x1E080
}

Returns code point corresponding to UTF-8 sequence of four bytes, where the first byte of the sequence is the
[byte1] and the others are in the [bytes] array starting from the [index]. Returns a non-positive value indicating
number of bytes from [bytes] included in malformed sequence if the sequence is malformed and
[throwOnMalformed] is false. @throws CharacterCodingException if the sequence of four bytes is
malformed and [throwOnMalformed] is true.

private fun codePointFrom4(bytes: ByteArray, byte1: Int, index:
Int, endIndex: Int, throwOnMalformed: Boolean): Int {
 if (index >= endIndex) {
 malformed(0, index,
throwOnMalformed)
 }
 val byte2 = bytes[index].toInt()
 if (byte1 and 0xF == 0x0) {
 if (byte2 and
0xF0 <= 0x80) {
 // Non-shortest form
 return malformed(0, index, throwOnMalformed)
 }
 } else if (byte1 and 0xF == 0x4) {
 if (byte2 and 0xF0 != 0x80) {
 // Out of Unicode code points
domain (larger than U+10FFFF)
 return malformed(0, index, throwOnMalformed)
 }
 } else if
(byte1 and 0xF > 0x4) {
 return malformed(0, index, throwOnMalformed)
 } else if (byte2
and 0xC0 != 0x80) {
 return malformed(0, index, throwOnMalformed)
 }
 if (index + 1 == endIndex)
 {
 return malformed(1, index, throwOnMalformed)
 }
 val byte3 = bytes[index + 1].toInt()
 if (byte3
and 0xC0 != 0x80) {
 return malformed(1, index, throwOnMalformed)
 }
 if (index + 2 == endIndex)
 {
 return malformed(2, index, throwOnMalformed)
 }
 val byte4 = bytes[index + 2].toInt()
 if (byte4
and 0xC0 != 0x80) {
 return malformed(2, index, throwOnMalformed)
 }
 return (byte1 shl 18) xor
(byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80
}

Maximum number of bytes needed to encode
a single char. Code points in `0..0x7F` are encoded in a single byte. Code points in `0x80..0x7FF` are
encoded in two bytes. Code points in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.
Surrogate code points in `0xD800..0xDFFF` are not Unicode scalar values, therefore
aren't encoded. Code points in `0x10000..0x10FFFF` are represented by a pair of surrogate `Char`s and are
encoded in four bytes.

private const val MAX_BYTES_PER_CHAR = 3
private val REPLACEMENT_BYTE_SEQUENCE:

```



ByteArray = byteArrayOf(0xEF.toByte(), 0xBF.toByte(), 0xBD.toByte())\n\n/\*\*\n \* Encodes the [string] using UTF-8 and returns the resulting [ByteArray].\n \* @param string the string to encode.\n \* @param startIndex the start offset (inclusive) of the substring to encode.\n \* @param endIndex the end offset (exclusive) of the substring to encode.\n \* @param throwOnMalformed whether to throw on malformed char sequence or replace by the [REPLACEMENT\_BYTE\_SEQUENCE].\n \* @throws CharacterCodingException if the char sequence is malformed and [throwOnMalformed] is true.\n \*/\ninternal fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): ByteArray

```
{\n require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)\n val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)\n var byteIndex = 0\n var charIndex = startIndex\n while (charIndex < endIndex) {\n val code = string[charIndex++].code\n when {\n code < 0x80 ->\n bytes[byteIndex++] = code.toByte()\n code < 0x800 -> {\n bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()\n bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n }\n code < 0xD800 || code >= 0xE000 -> {\n bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()\n bytes[byteIndex++] = (((code shr 6) and 0x3F) or 0x80).toByte()\n bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n }\n else -> { // Surrogate char value\n val codePoint = codePointFromSurrogate(string, code, charIndex, endIndex, throwOnMalformed)\n if (codePoint <= 0) {\n bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[0]\n bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[1]\n bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[2]\n } else {\n bytes[byteIndex++] = ((codePoint shr 18) or 0xF0).toByte()\n bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()\n bytes[byteIndex++] = (((codePoint shr 6) and 0x3F) or 0x80).toByte()\n bytes[byteIndex++] = ((codePoint and 0x3F) or 0x80).toByte()\n }\n charIndex++\n }\n }\n }\n return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)\n}\n\n/**\n * The character a malformed UTF-8 byte sequence is replaced by.\n */\nprivate const val REPLACEMENT_CHAR = "\uFFFF"\n\n/**\n * Decodes the UTF-8 [bytes] array and returns
```

```
the resulting [String].\n * @param bytes the byte array to decode.\n * @param startIndex the start offset (inclusive) of the array to be decoded.\n * @param endIndex the end offset (exclusive) of the array to be encoded.\n * @param throwOnMalformed whether to throw on malformed byte sequence or replace by the [REPLACEMENT_CHAR].\n * @throws CharacterCodingException if the array is malformed UTF-8 byte sequence and [throwOnMalformed] is true.\n */\ninternal fun decodeUtf8(bytes: ByteArray, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {\n require(startIndex >= 0 && endIndex <= bytes.size && startIndex <= endIndex)\n var byteIndex = startIndex\n val stringBuilder = StringBuilder()\n while (byteIndex < endIndex) {\n val byte = bytes[byteIndex++].toInt()\n when {\n byte >= 0 ->\n stringBuilder.append(byte.toChar())\n byte shr 5 == -2 -> {\n val code = codePointFrom2(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n if (code <= 0) {\n stringBuilder.append(REPLACEMENT_CHAR)\n byteIndex += -code\n } else {\n stringBuilder.append(code.toChar())\n byteIndex += 1\n }\n }\n byte shr 4 == -2 -> {\n val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n if (code <= 0) {\n stringBuilder.append(REPLACEMENT_CHAR)\n byteIndex += -code\n } else {\n stringBuilder.append(code.toChar())\n byteIndex += 2\n }\n }\n byte shr 3 == -2 -> {\n val code = codePointFrom4(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n if (code <= 0) {\n stringBuilder.append(REPLACEMENT_CHAR)\n byteIndex += -code\n } else {\n val high = (code - 0x10000) shr 10 or 0xD800\n val low = (code and 0x3FF) or 0xDC00\n stringBuilder.append(high.toChar())\n stringBuilder.append(low.toChar())\n byteIndex += 3\n }\n }\n else -> {\n
```

```

malformed(0, byteIndex, throwOnMalformed)\n stringBuilder.append(REPLACEMENT_CHAR)\n
}\n }\n }\n\n return stringBuilder.toString()\n}", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *^\n\npackage kotlin\n\n/**\n * Returns the detailed description of this
throwable with its stack trace.\n *^\n * The detailed description includes:\n * - the short description (see
[Throwable.toString]) of this throwable;\n * - the complete
stack trace;\n * - detailed descriptions of the exceptions that were [suppressed][suppressedExceptions] in order to
deliver this exception;\n * - the detailed description of each throwable in the [Throwable.cause] chain.\n
*^\n\n@SinceKotlin("1.4")\npublic actual fun Throwable.stackTraceToString(): String =
ExceptionTraceBuilder().buildFor(this)\n\n/**\n * Prints the [detailed description][Throwable.stackTraceToString]
of this throwable to console error output.\n *^\n\n@SinceKotlin("1.4")\npublic actual fun
Throwable.printStackTrace() {\n console.error(this.stackTraceToString())\n}\n\n/**\n * Adds the specified
exception to the list of exceptions that were\n * suppressed in order to deliver this exception.\n
*^\n\n@SinceKotlin("1.4")\npublic actual fun Throwable.addSuppressed(exception: Throwable) {\n if (this !==
exception) {\n val suppressed = this.asDynamic()._suppressed.unsafeCast<MutableList<Throwable>?>()\n
if (suppressed == null) {\n this.asDynamic()._suppressed
= mutableListOf(exception)\n } else {\n suppressed.add(exception)\n }\n }\n}\n\n/**\n * Returns
a list of all exceptions that were suppressed in order to deliver this exception.\n *^\n\n@SinceKotlin("1.4")\npublic
actual val Throwable.suppressedExceptions: List<Throwable>\n get() {\n return
this.asDynamic()._suppressed?.unsafeCast<List<Throwable>>() ?: emptyList()\n }\n\nprivate class
ExceptionTraceBuilder {\n private val target = StringBuilder()\n private val visited = arrayOf<Throwable>()\n
private var topStack: String = ""\n private var topStackStart: Int = 0\n fun buildFor(exception: Throwable):
String {\n exception.dumpFullTrace("", "")\n return target.toString()\n }\n private fun
hasSeen(exception: Throwable): Boolean = visited.any { it === exception }\n private fun
Throwable.dumpFullTrace(indent: String, qualifier: String) {\n this.dumpSelfTrace(indent, qualifier)
|| return\n var cause = this.cause\n while (cause != null) {\n cause.dumpSelfTrace(indent, "Caused
by: ") || return\n cause = cause.cause\n }\n }\n private fun Throwable.dumpSelfTrace(indent:
String, qualifier: String): Boolean {\n target.append(indent).append(qualifier)\n val shortInfo =
this.toString()\n if (hasSeen(this)) {\n target.append("[CIRCULAR REFERENCE, SEE ABOVE:
\").append(shortInfo).append("]\n")\n return false\n }\n visited.asDynamic().push(this)\n var
stack = this.asDynamic().stack as String?\n if (stack != null) {\n val stackStart =
stack.indexOf(shortInfo).let { if (it < 0) 0 else it + shortInfo.length }\n if (stackStart == 0)
target.append(shortInfo).append("\n")\n if (topStack.isEmpty()) {\n topStack = stack\n
topStackStart = stackStart\n } else {\n
 stack = dropCommonFrames(stack, stackStart)\n }\n if (indent.isNotEmpty()) {\n //
indent stack, but avoid indenting exception message lines\n val messageLines = if (stackStart == 0) 0 else
1 + shortInfo.count { c -> c == '\n' }\n stack.lineSequence().forEachIndexed { index: Int, line: String ->}\n
 if (index >= messageLines) target.append(indent)\n target.append(line).append("\n")\n
 }\n } else {\n target.append(stack).append("\n")\n } } else {\n
target.append(shortInfo).append("\n")\n }\n val suppressed = suppressedExceptions\n if
(suppressed.isNotEmpty()) {\n val suppressedIndent = indent + " "\n for (s in suppressed) {\n
 s.dumpFullTrace(suppressedIndent, "Suppressed: ")\n }\n }\n return true\n }\n
private fun dropCommonFrames(stack: String, stackStart: Int): String {\n var commonFrames: Int = 0\n
var lastBreak: Int = 0\n var preLastBreak: Int = 0\n for (pos in 0 until minOf(topStack.length -
topStackStart, stack.length - stackStart)) {\n val c = stack[stack.lastIndex - pos]\n if (c !=
topStack[topStack.lastIndex - pos]) break\n if (c == '\n') {\n commonFrames += 1\n
preLastBreak = lastBreak\n lastBreak = pos\n }\n }\n if (commonFrames <= 1) return
stack\n while (preLastBreak > 0 && stack[stack.lastIndex - (preLastBreak - 1)] == ')'\n preLastBreak -=

```

```

1\n\n // leave 1 common frame to ease matching with the top exception stack\n return
stack.dropLast(preLastBreak) + "\"... and ${commonFrames - 1} more common stack frames skipped\"\n
}\n}\",\"/>\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.js.json\nimport kotlin.math.*\n\ninternal
actual inline val durationAssertionsEnabled: Boolean get() = true\n\ninternal actual fun
formatToExactDecimals(value: Double, decimals: Int): String {\n val rounded = if (decimals == 0) {\n value\n
} else {\n val pow = 10.0.pow(decimals)\n JsMath.round(abs(value) * pow) / pow * sign(value)\n }\n
return if (abs(rounded) < 1e21) {\n // toFixed switches to scientific format after 1e21\n
rounded.asDynamic().toFixed(decimals).unsafeCast<String>()\n } else {\n // toPrecision outputs the specified
number of digits, but only for positive numbers\n val positive = abs(rounded)\n val positiveString =
positive.asDynamic().toPrecision(ceil(log10(positive)) + decimals).unsafeCast<String>()\n if (rounded < 0)
\"-$positiveString\" else positiveString\n }\n}\n\ninternal actual fun formatUpToDecimals(value: Double,
decimals: Int): String {\n return value.asDynamic().toLocaleString(\"en-us\", json(\"maximumFractionDigits\" to
decimals)).unsafeCast<String>()\n}\n}\",\"/>\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.time\n\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic actual enum class
DurationUnit(internal val scale: Double) {\n /**\n * Time unit representing one nanosecond, which is 1/1000
of a microsecond.\n */\n NANOSECONDS(1e0),\n /**\n * Time unit representing one microsecond, which is
1/1000 of a millisecond.\n */\n MICROSECONDS(1e3),\n /**\n * Time unit representing one millisecond,
which is 1/1000 of a second.\n */\n MILLISECONDS(1e6),\n
/**\n * Time unit representing one second.\n */\n SECONDS(1e9),\n /**\n * Time unit representing
one minute.\n */\n MINUTES(60e9),\n /**\n * Time unit representing one hour.\n */\n
HOURS(3600e9),\n /**\n * Time unit representing one day, which is always equal to 24 hours.\n */\n
DAYS(86400e9);\n}\n}\n\n@SinceKotlin(\"1.3\")\ninternal actual fun convertDurationUnit(value: Double, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Double {\n val sourceCompareTarget =
sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> value *
(sourceUnit.scale / targetUnit.scale)\n sourceCompareTarget < 0 -> value / (targetUnit.scale /
sourceUnit.scale)\n else -> value\n }\n}\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun
convertDurationUnitOverflow(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n val
sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n
return when {\n sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale).toLong()\n
sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n else -> value\n
}\n}\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun convertDurationUnit(value: Long, sourceUnit: DurationUnit,
targetUnit: DurationUnit): Long {\n val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n
return when {\n sourceCompareTarget > 0 -> {\n val scale = (sourceUnit.scale /
targetUnit.scale).toLong()\n val result = value * scale\n when {\n result / scale == value ->
result\n value > 0 -> Long.MAX_VALUE\n else -> Long.MIN_VALUE\n }\n }\n
sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n else -> value\n
}\n}\n}\n}\",\"/>\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport
org.w3c.performance.GlobalPerformance\nimport org.w3c.performance.Performance\nimport
kotlin.math.truncate\nimport kotlin.time.Duration.Companion.milliseconds\nimport
kotlin.time.TimeSource.Monotonic.ValueTimeMark\n\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\") //
visibility\n\ninternal actual typealias ValueTimeMarkReading = Any\n\n@ExperimentalTime\n\ninternal interface
DefaultTimeSource : TimeSource.WithComparableMarks {\n override fun markNow(): ValueTimeMark\n fun

```

```

elapsedFrom(timeMark: ValueTimeMark): Duration\n fun differenceBetween(one: ValueTimeMark, another:
ValueTimeMark): Duration\n fun adjustReading(timeMark: ValueTimeMark, duration: Duration):
ValueTimeMark\n}\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal actual object MonotonicTimeSource
: DefaultTimeSource, TimeSource.WithComparableMarks
{ // TODO: interface should not be required here\n\n private val actualSource: DefaultTimeSource = run {\n
val isNode: Boolean = js("typeof process !== 'undefined' && process.versions && !process.versions.node")\n\n
if (isNode)\n HrTimeSource(js("process").unsafeCast<Process>())\n else\n js("typeof self !==
'undefined' ? self : globalThis")\n .unsafeCast<GlobalPerformance?>()\n ?.performance\n
?.let(::PerformanceTimeSource)\n ?: DateNowTimeSource\n }\n\n actual override fun markNow():
ValueTimeMark = actualSource.markNow()\n actual override fun elapsedFrom(timeMark: ValueTimeMark):
Duration = actualSource.elapsedFrom(timeMark)\n actual override fun differenceBetween(one: ValueTimeMark,
another: ValueTimeMark): Duration = actualSource.differenceBetween(one, another)\n\n actual override fun
adjustReading(timeMark: ValueTimeMark, duration: Duration):
ValueTimeMark =\n actualSource.adjustReading(timeMark, duration)\n}\n\ninternal external interface Process
{\n fun hrtime(time: Array<Double> = definedExternally):
Array<Double>\n}\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal class HrTimeSource(private val
process: Process) : DefaultTimeSource {\n\n override fun markNow(): ValueTimeMark =
ValueTimeMark(process.hrtime())\n override fun elapsedFrom(timeMark: ValueTimeMark): Duration =\n
@Suppress("UNCHECKED_CAST")\n process.hrtime(timeMark.reading as Array<Double>)\n .let {
(seconds, nanos) -> seconds.toDuration(DurationUnit.SECONDS) +
nanos.toDuration(DurationUnit.NANOSECONDS) }\n\n @Suppress("UNCHECKED_CAST")\n override fun
differenceBetween(one: ValueTimeMark, another: ValueTimeMark): Duration {\n val (s1, n1) = one.reading as
Array<Double>\n val (s2, n2) = another.reading as Array<Double>\n return (if (s1 == s2 && n1 == n2)
Duration.ZERO else (s1 - s2).toDuration(DurationUnit.SECONDS))
+ (n1 - n2).toDuration(DurationUnit.NANOSECONDS)\n }\n\n override fun adjustReading(timeMark:
ValueTimeMark, duration: Duration): ValueTimeMark =\n @Suppress("UNCHECKED_CAST")\n
(timeMark.reading as Array<Double>).let { (seconds, nanos) ->\n duration.toComponents { _, addNanos ->\n
val resultSeconds = sumCheckNaN(seconds + truncate(duration.toDouble(DurationUnit.SECONDS)))\n
arrayOf<Double>(resultSeconds, if (resultSeconds.isFinite()) nanos + addNanos else 0.0)\n }\n
}.let(TimeSource.Monotonic::ValueTimeMark)\n}\n\n override fun toString(): String =
"TimeSource(process.hrtime())"\n}\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal class
PerformanceTimeSource(val performance: Performance) : DefaultTimeSource { //
AbstractDoubleTimeSource(unit = DurationUnit.MILLISECONDS) {\n private fun read(): Double =
performance.now()\n\n override fun markNow(): ValueTimeMark
= ValueTimeMark(read())\n override fun elapsedFrom(timeMark: ValueTimeMark): Duration = (read() -
timeMark.reading as Double).milliseconds\n\n override fun differenceBetween(one: ValueTimeMark, another:
ValueTimeMark): Duration {\n val ms1 = one.reading as Double\n val ms2 = another.reading as Double\n
return if (ms1 == ms2) Duration.ZERO else (ms1 - ms2).milliseconds\n }\n\n override fun
adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +
duration.toDouble(DurationUnit.MILLISECONDS)))\n\n override fun toString(): String =
"TimeSource(self.performance.now())"\n}\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal object
DateNowTimeSource : DefaultTimeSource {\n private fun read(): Double = kotlin.js.Date.now()\n\n override
fun markNow(): ValueTimeMark = ValueTimeMark(read())\n override fun elapsedFrom(timeMark:
ValueTimeMark): Duration = (read()
- timeMark.reading as Double).milliseconds\n\n override fun differenceBetween(one: ValueTimeMark, another:
ValueTimeMark): Duration {\n val ms1 = one.reading as Double\n val ms2 = another.reading as Double\n
return if (ms1 == ms2) Duration.ZERO else (ms1 - ms2).milliseconds\n }\n\n override fun

```

```

adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +
duration.toDouble(DurationUnit.MILLISECONDS)))\n\n override fun toString(): String =
`\"TimeSource(Date.now())\"`\n\nprivate fun sumCheckNaN(value: Double): Double = value.also { if (it.isNaN())
throw IllegalArgumentException(`\"Summing infinities of different signs`) }\", \"/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport
org.w3c.dom.*\n\nimport
kotlin.contracts.*\n\n/**\n * Creates a new element with the specified [name].\n * The element is initialized
with the specified [init] function.\n */\n\n@SinceKotlin(\"1.4\")\npublic fun Document.createElement(name: String,
init: Element.() -> Unit): Element {\n contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }\n return createElement(name).apply(init)\n}\n\n/**\n * Appends a newly created element with the specified [name] to
this element.\n * The element is initialized with the specified [init] function.\n */\n\n@SinceKotlin(\"1.4\")\npublic fun Element.appendChild(name: String, init: Element.() -> Unit): Element {\n contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }\n return
ownerDocument!!.createElement(name, init).also { appendChild(it) }\n}\n\n\", \"/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt
file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/**\n * Returns true if the element has the given CSS
class style in its 'class' attribute\n */\n\n@SinceKotlin(\"1.4\")\nfun Element.hasClass(cssClass: String): Boolean =
className.matches(`\"\"(\\^(.*)\\$cssClass(\\$|\\s+\\.*)\"\".toRegex())`\n\n/**\n * Adds CSS class to element. Has no
effect if all specified classes are already in class attribute of the element\n */\n * @return true if at least one class has
been added\n */\n\n@SinceKotlin(\"1.4\")\nfun Element.addClass(vararg cssClasses: String): Boolean {\n val
missingClasses = cssClasses.filterNot { hasClass(it) }\n if (missingClasses.isNotEmpty()) {\n val
presentClasses = className.trim()\n className = buildString {\n append(presentClasses)\n if
(!presentClasses.isEmpty()) {\n append(\" \")\n }\n missingClasses.joinTo(this, \" \")\n }\n return true\n }\n return false\n}\n\n/**\n * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the
element\n */\n * @return true if at least one class has been removed\n */\n\n@SinceKotlin(\"1.4\")\nfun
Element.removeClass(vararg cssClasses: String): Boolean {\n if (cssClasses.any { hasClass(it) }) {\n val
toBeRemoved = cssClasses.toSet()\n className = className.trim().split(`\"\"\\s+\"\".toRegex()).filter { it !in
toBeRemoved }.joinToString(\" \")\n return true\n }\n return false\n}\n\n\", \"/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage
kotlin.text\n\n/**\n * Converts the string into a regular expression [Regex] with the default options.\n */\n\n@kotlin.internal.InlineOnly\npublic inline
fun String.toRegex(): Regex = Regex(this)\n\n/**\n * Converts the string into a regular expression [Regex] with the
specified single [option].\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun String.toRegex(option: RegexOptions):
Regex = Regex(this, option)\n\n/**\n * Converts the string into a regular expression [Regex] with the specified set
of [options].\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun String.toRegex(options: Set<RegexOption>):
Regex = Regex(this, options)\n\n\", \"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/**\n * Gets a value indicating
whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.\n */\n\n@SinceKotlin(\"1.4\")\npublic val
Node.isText: Boolean\n get() = nodeType == Node.TEXT_NODE || nodeType ==
Node.CDATA_SECTION_NODE\n\n/**\n *

```

```

Gets a value indicating whether this node is an [Element].\n *\n@SinceKotlin("1.4")\npublic val Node.isElement:
Boolean\n get() = nodeType == Node.ELEMENT_NODE\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/**
Removes all the children from this node. *\n@SinceKotlin("1.4")\npublic fun Node.clear() {\n while
(hasChildNodes()) {\n removeChild(firstChild!!)\n }\n}\n\n/**\n * Creates text node and append it to the
element.\n *\n * @return this element\n *\n@SinceKotlin("1.4")\nfun Element.appendText(text: String): Element
{\n appendChild(ownerDocument!!.createTextNode(text))\n return this\n}\n"/*\n * Copyright 2010-2019
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n *\n\npackage org.w3c.dom\n\n@Deprecated("Use
UnionMessagePortOrWindowProxy instead.", ReplaceWith("UnionMessagePortOrWindowProxy"))\ntypealias
UnionMessagePortOrWindow = UnionMessagePortOrWindowProxy\n\n@Deprecated("Use `as` instead.",
ReplaceWith("`as`"))\nvar HTMLLinkElement.as_ get() = `as`\n set(value) {\n `as` = value\n }\n\n@Deprecated("Use `is` instead.", ReplaceWith("`is`"))\nvar ElementCreationOptions.is_ get() = `is`\n
set(value) {\n `is` = value\n }"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See
github.com/kotlin/dukat for details\n\npackage org.khronos.webgl\n\nimport kotlin.js.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external
interface WebGLContextAttributes {\n var alpha: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var depth: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var stencil: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var antialias: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var premultipliedAlpha: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n var preserveDrawingBuffer: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var preferLowPowerToHighPerformance: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var failIfMajorPerformanceCaveat:
Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\npublic inline fun WebGLContextAttributes(alpha:
Boolean? = true, depth: Boolean? = true, stencil: Boolean? = false, antialias: Boolean? = true, premultipliedAlpha:
Boolean? = true, preserveDrawingBuffer: Boolean? = false, preferLowPowerToHighPerformance: Boolean? = false,
failIfMajorPerformanceCaveat: Boolean? = false): WebGLContextAttributes {\n val o = js("{}")\n
o["alpha"] = alpha\n o["depth"] = depth\n o["stencil"] = stencil\n o["antialias"] = antialias\n
o["premultipliedAlpha"] = premultipliedAlpha\n o["preserveDrawingBuffer"] = preserveDrawingBuffer\n
o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance\n
o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat\n return o\n}\n\npublic external abstract
class WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer)
to Kotlin\n *\n\npublic external abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin\n *\n\npublic
external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin\n *\n\npublic external
abstract class WebGLProgram : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin\n *\n\npublic
external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin\n *\n\npublic external

```

```

abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin\n *\npublic
external abstract class WebGLTexture : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin\n
*\npublic external abstract class WebGLUniformLocation\n\n/**\n * Exposes the JavaScript
[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin\n *\npublic
external abstract class WebGLActiveInfo {\n open val size: Int\n open val type: Int\n open val name:
String\n}\n\n/**\n * Exposes the JavaScript
[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to
Kotlin\n *\npublic external abstract class WebGLShaderPrecisionFormat {\n open val rangeMin: Int\n open val
rangeMax: Int\n open val precision:
Int\n}\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface
WebGLRenderingContextBase {\n val canvas: HTMLCanvasElement\n val drawingBufferWidth: Int\n val
drawingBufferHeight:
Int\n fun getContextAttributes(): WebGLContextAttributes?\n fun isContextLost(): Boolean\n fun
getSupportedExtensions(): Array<String>?\n fun getExtension(name: String): dynamic\n fun
activeTexture(texture: Int)\n fun attachShader(program: WebGLProgram?, shader: WebGLShader?)\n fun
bindAttribLocation(program: WebGLProgram?, index: Int, name: String)\n fun bindBuffer(target: Int, buffer:
WebGLBuffer?)\n fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)\n fun
bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)\n fun bindTexture(target: Int, texture:
WebGLTexture?)\n fun blendColor(red: Float, green: Float, blue: Float, alpha: Float)\n fun
blendEquation(mode: Int)\n fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)\n fun
blendFunc(sfactor: Int, dfactor: Int)\n fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha:
Int)\n fun bufferData(target: Int, size: Int, usage: Int)\n
 fun bufferData(target: Int, data: BufferDataSource?, usage: Int)\n fun bufferSubData(target: Int, offset: Int, data:
BufferDataSource?)\n fun checkFramebufferStatus(target: Int): Int\n fun clear(mask: Int)\n fun clearColor(red:
Float, green: Float, blue: Float, alpha: Float)\n fun clearDepth(depth: Float)\n fun clearStencil(s: Int)\n fun
colorMask(red: Boolean, green: Boolean, blue: Boolean, alpha: Boolean)\n fun compileShader(shader:
WebGLShader?)\n fun compressedTexImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int,
border: Int, data: ArrayBufferView)\n fun compressedTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset:
Int, width: Int, height: Int, format: Int, data: ArrayBufferView)\n fun copyTexImage2D(target: Int, level: Int,
internalformat: Int, x: Int, y: Int, width: Int, height: Int, border: Int)\n fun copyTexSubImage2D(target: Int, level:
Int, xoffset: Int, yoffset: Int, x: Int, y: Int, width: Int, height:
Int)\n fun createBuffer(): WebGLBuffer?\n fun createFramebuffer(): WebGLFramebuffer?\n fun
createProgram(): WebGLProgram?\n fun createRenderbuffer(): WebGLRenderbuffer?\n fun createShader(type:
Int): WebGLShader?\n fun createTexture(): WebGLTexture?\n fun cullFace(mode: Int)\n fun
deleteBuffer(buffer: WebGLBuffer?)\n fun deleteFramebuffer(framebuffer: WebGLFramebuffer?)\n fun
deleteProgram(program: WebGLProgram?)\n fun deleteRenderbuffer(renderbuffer: WebGLRenderbuffer?)\n fun
deleteShader(shader: WebGLShader?)\n fun deleteTexture(texture: WebGLTexture?)\n fun depthFunc(func:
Int)\n fun depthMask(flag: Boolean)\n fun depthRange(zNear: Float, zFar: Float)\n fun detachShader(program:
WebGLProgram?, shader: WebGLShader?)\n fun disable(cap: Int)\n fun disableVertexAttribArray(index: Int)\n fun
drawArrays(mode: Int, first: Int, count: Int)\n fun drawElements(mode: Int, count: Int, type: Int, offset: Int)\n
fun enable(cap:
Int)\n fun enableVertexAttribArray(index: Int)\n fun finish()\n fun flush()\n fun
framebufferRenderbuffer(target: Int, attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)\n
fun framebufferTexture2D(target: Int, attachment: Int, textarget: Int, texture: WebGLTexture?, level: Int)\n fun
frontFace(mode: Int)\n fun generateMipmap(target: Int)\n fun getActiveAttrib(program: WebGLProgram?,
index: Int): WebGLActiveInfo?\n fun getActiveUniform(program: WebGLProgram?, index: Int):

```

```

WebGLActiveInfo? fun getAttachedShaders(program: WebGLProgram?): Array<WebGLShader?> fun
getAttribLocation(program: WebGLProgram?, name: String): Int fun getBufferParameter(target: Int, pname:
Int): Any? fun getParameter(pname: Int): Any? fun getError(): Int fun
getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Any? fun
getProgramParameter(program: WebGLProgram?, pname: Int): Any? fun getProgramInfoLog(program:
WebGLProgram?): String? fun getRenderbufferParameter(target: Int, pname: Int): Any? fun
getShaderParameter(shader: WebGLShader?, pname: Int): Any? fun getShaderPrecisionFormat(shaderType: Int,
precisionType: Int): WebGLShaderPrecisionFormat? fun getShaderInfoLog(shader: WebGLShader?): String?
fun getShaderSource(shader: WebGLShader?): String? fun getTexParameter(target: Int, pname: Int): Any?
fun getUniform(program: WebGLProgram?, location: WebGLUniformLocation?): Any? fun
getUniformLocation(program: WebGLProgram?, name: String): WebGLUniformLocation? fun
getVertexAttrib(index: Int, pname: Int): Any? fun getVertexAttribOffset(index: Int, pname: Int): Int fun
hint(target: Int, mode: Int) fun isBuffer(buffer: WebGLBuffer?): Boolean fun isEnabled(cap: Int): Boolean
fun isFramebuffer(framebuffer: WebGLFramebuffer?): Boolean fun isProgram(program: WebGLProgram?):
Boolean fun isRenderbuffer(renderbuffer:
WebGLRenderbuffer?): Boolean fun isShader(shader: WebGLShader?): Boolean fun isTexture(texture:
WebGLTexture?): Boolean fun lineWidth(width: Float) fun linkProgram(program: WebGLProgram?)
fun pixelStorei(pname: Int, param: Int) fun polygonOffset(factor: Float, units: Float) fun readPixels(x: Int, y:
Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?) fun renderbufferStorage(target: Int,
internalformat: Int, width: Int, height: Int) fun sampleCoverage(value: Float, invert: Boolean) fun scissor(x:
Int, y: Int, width: Int, height: Int) fun shaderSource(shader: WebGLShader?, source: String) fun
stencilFunc(func: Int, ref: Int, mask: Int) fun stencilFuncSeparate(face: Int, func: Int, ref: Int, mask: Int) fun
stencilMask(mask: Int) fun stencilMaskSeparate(face: Int, mask: Int) fun stencilOp(fail: Int, zfail: Int, zpass:
Int) fun stencilOpSeparate(face: Int, fail: Int, zfail: Int,
zpass: Int) fun texImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, format:
Int, type: Int, pixels: ArrayBufferView?) fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int,
type: Int, source: TexImageSource?) fun texParameterf(target: Int, pname: Int, param: Float) fun
texParameteri(target: Int, pname: Int, param: Int) fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset:
Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?) fun texSubImage2D(target: Int,
level: Int, xoffset: Int, yoffset: Int, format: Int, type: Int, source: TexImageSource?) fun uniform1f(location:
WebGLUniformLocation?, x: Float) fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform1fv(location: WebGLUniformLocation?, v: Array<Float>) fun uniform1i(location:
WebGLUniformLocation?, x: Int) fun uniform1iv(location: WebGLUniformLocation?, v:
Int32Array) fun uniform1iv(location: WebGLUniformLocation?, v: Array<Int>) fun uniform2f(location:
WebGLUniformLocation?, x: Float, y: Float) fun uniform2fv(location: WebGLUniformLocation?, v:
Float32Array) fun uniform2fv(location: WebGLUniformLocation?, v: Array<Float>) fun
uniform2i(location: WebGLUniformLocation?, x: Int, y: Int) fun uniform2iv(location:
WebGLUniformLocation?, v: Int32Array) fun uniform2iv(location: WebGLUniformLocation?, v: Array<Int>)
fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float) fun uniform3fv(location:
WebGLUniformLocation?, v: Float32Array) fun uniform3fv(location: WebGLUniformLocation?, v:
Array<Float>) fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int) fun
uniform3iv(location: WebGLUniformLocation?, v: Int32Array) fun uniform3iv(location:
WebGLUniformLocation?, v: Array<Int>) fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float, w: Float) fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array) fun
uniform4fv(location: WebGLUniformLocation?, v: Array<Float>) fun uniform4i(location:
WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int) fun uniform4iv(location: WebGLUniformLocation?, v:
Int32Array) fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>) fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array) fun

```



```

uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
useProgram(program:
WebGLProgram?)\n fun validateProgram(program: WebGLProgram?)\n fun vertexAttrib1f(index: Int, x:
Float)\n fun vertexAttrib1fv(index: Int, values: dynamic)\n fun vertexAttrib2f(index: Int, x: Float, y: Float)\n
fun vertexAttrib2fv(index: Int, values: dynamic)\n fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)\n fun
vertexAttrib3fv(index: Int, values: dynamic)\n fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)\n
fun vertexAttrib4fv(index: Int, values: dynamic)\n fun vertexAttribPointer(index: Int, size: Int, type: Int,
normalized: Boolean, stride: Int, offset: Int)\n fun viewport(x: Int, y: Int, width: Int, height: Int)\n\n companion
object {\n val DEPTH_BUFFER_BIT: Int\n val STENCIL_BUFFER_BIT: Int\n val
COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES: Int\n val LINE_LOOP: Int\n val
LINE_STRIP: Int\n val TRIANGLES: Int\n val TRIANGLE_STRIP:
Int\n val TRIANGLE_FAN: Int\n val ZERO: Int\n val ONE: Int\n val SRC_COLOR: Int\n
val ONE_MINUS_SRC_COLOR: Int\n val SRC_ALPHA: Int\n val ONE_MINUS_SRC_ALPHA: Int\n
val DST_ALPHA: Int\n val ONE_MINUS_DST_ALPHA: Int\n val DST_COLOR: Int\n val
ONE_MINUS_DST_COLOR: Int\n val SRC_ALPHA_SATURATE: Int\n val FUNC_ADD: Int\n val
BLEND_EQUATION: Int\n val BLEND_EQUATION_RGB: Int\n val BLEND_EQUATION_ALPHA:
Int\n val FUNC_SUBTRACT: Int\n val FUNC_REVERSE_SUBTRACT: Int\n val
BLEND_DST_RGB: Int\n val BLEND_SRC_RGB: Int\n val BLEND_DST_ALPHA: Int\n val
BLEND_SRC_ALPHA: Int\n val CONSTANT_COLOR: Int\n val ONE_MINUS_CONSTANT_COLOR:
Int\n val CONSTANT_ALPHA: Int\n val ONE_MINUS_CONSTANT_ALPHA: Int\n val
BLEND_COLOR: Int\n val ARRAY_BUFFER: Int\n val ELEMENT_ARRAY_BUFFER: Int\n val
ARRAY_BUFFER_BINDING:
Int\n val ELEMENT_ARRAY_BUFFER_BINDING: Int\n val STREAM_DRAW: Int\n val
STATIC_DRAW: Int\n val DYNAMIC_DRAW: Int\n val BUFFER_SIZE: Int\n val
BUFFER_USAGE: Int\n val CURRENT_VERTEX_ATTRIB: Int\n val FRONT: Int\n val BACK:
Int\n val FRONT_AND_BACK: Int\n val CULL_FACE: Int\n val BLEND: Int\n val DITHER:
Int\n val STENCIL_TEST: Int\n val DEPTH_TEST: Int\n val SCISSOR_TEST: Int\n val
POLYGON_OFFSET_FILL: Int\n val SAMPLE_ALPHA_TO_COVERAGE: Int\n val
SAMPLE_COVERAGE: Int\n val NO_ERROR: Int\n val INVALID_ENUM: Int\n val
INVALID_VALUE: Int\n val INVALID_OPERATION: Int\n val OUT_OF_MEMORY: Int\n val CW:
Int\n val CCW: Int\n val LINE_WIDTH: Int\n val ALIASED_POINT_SIZE_RANGE: Int\n val
ALIASED_LINE_WIDTH_RANGE: Int\n val CULL_FACE_MODE: Int\n val FRONT_FACE: Int\n
val DEPTH_RANGE: Int\n val DEPTH_WRITEMASK: Int\n val DEPTH_CLEAR_VALUE: Int\n
val DEPTH_FUNC: Int\n val STENCIL_CLEAR_VALUE: Int\n val STENCIL_FUNC: Int\n val
STENCIL_FAIL: Int\n val STENCIL_PASS_DEPTH_FAIL: Int\n val STENCIL_PASS_DEPTH_PASS:
Int\n val STENCIL_REF: Int\n val STENCIL_VALUE_MASK: Int\n val STENCIL_WRITEMASK:
Int\n val STENCIL_BACK_FUNC: Int\n val STENCIL_BACK_FAIL: Int\n val
STENCIL_BACK_PASS_DEPTH_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_PASS: Int\n val
STENCIL_BACK_REF: Int\n val STENCIL_BACK_VALUE_MASK: Int\n val
STENCIL_BACK_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR_BOX: Int\n val
COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n
val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n
val SUBPIXEL_BITS: Int\n

```

val RED\_BITS: Int\n     val GREEN\_BITS: Int\n     val BLUE\_BITS: Int\n     val ALPHA\_BITS: Int\n  
 val DEPTH\_BITS: Int\n     val STENCIL\_BITS: Int\n     val POLYGON\_OFFSET\_UNITS: Int\n     val  
 POLYGON\_OFFSET\_FACTOR: Int\n     val TEXTURE\_BINDING\_2D: Int\n     val SAMPLE\_BUFFERS:  
 Int\n     val SAMPLES: Int\n     val SAMPLE\_COVERAGE\_VALUE: Int\n     val  
 SAMPLE\_COVERAGE\_INVERT: Int\n     val COMPRESSED\_TEXTURE\_FORMATS: Int\n     val  
 DONT\_CARE: Int\n     val FASTEST: Int\n     val NICEST: Int\n     val GENERATE\_MIPMAP\_HINT: Int\n  
    val BYTE: Int\n     val UNSIGNED\_BYTE: Int\n     val SHORT: Int\n     val UNSIGNED\_SHORT: Int\n  
    val INT: Int\n     val UNSIGNED\_INT: Int\n     val FLOAT: Int\n     val DEPTH\_COMPONENT: Int\n  
 val ALPHA: Int\n     val RGB: Int\n     val RGBA: Int\n     val LUMINANCE: Int\n     val  
 LUMINANCE\_ALPHA: Int\n     val UNSIGNED\_SHORT\_4\_4\_4\_4: Int\n     val  
 UNSIGNED\_SHORT\_5\_5\_5\_1:  
 Int\n     val UNSIGNED\_SHORT\_5\_6\_5: Int\n     val FRAGMENT\_SHADER: Int\n     val  
 VERTEX\_SHADER: Int\n     val MAX\_VERTEX\_ATTRIBS: Int\n     val  
 MAX\_VERTEX\_UNIFORM\_VECTORS: Int\n     val MAX\_VARYING\_VECTORS: Int\n     val  
 MAX\_COMBINED\_TEXTURE\_IMAGE\_UNITS: Int\n     val MAX\_VERTEX\_TEXTURE\_IMAGE\_UNITS:  
 Int\n     val MAX\_TEXTURE\_IMAGE\_UNITS: Int\n     val MAX\_FRAGMENT\_UNIFORM\_VECTORS: Int\n  
    val SHADER\_TYPE: Int\n     val DELETE\_STATUS: Int\n     val LINK\_STATUS: Int\n     val  
 VALIDATE\_STATUS: Int\n     val ATTACHED\_SHADERS: Int\n     val ACTIVE\_UNIFORMS: Int\n     val  
 ACTIVE\_ATTRIBUTES: Int\n     val SHADING\_LANGUAGE\_VERSION: Int\n     val  
 CURRENT\_PROGRAM: Int\n     val NEVER: Int\n     val LESS: Int\n     val EQUAL: Int\n     val  
 LEQUAL: Int\n     val GREATER: Int\n     val NOTEQUAL: Int\n     val GEQUAL: Int\n     val ALWAYS:  
 Int\n     val KEEP: Int\n     val REPLACE: Int\n     val  
 INCR: Int\n     val DECR: Int\n     val INVERT: Int\n     val INCR\_WRAP: Int\n     val DECR\_WRAP: Int\n  
    val VENDOR: Int\n     val RENDERER: Int\n     val VERSION: Int\n     val NEAREST: Int\n     val  
 LINEAR: Int\n     val NEAREST\_MIPMAP\_NEAREST: Int\n     val LINEAR\_MIPMAP\_NEAREST: Int\n  
 val NEAREST\_MIPMAP\_LINEAR: Int\n     val LINEAR\_MIPMAP\_LINEAR: Int\n     val  
 TEXTURE\_MAG\_FILTER: Int\n     val TEXTURE\_MIN\_FILTER: Int\n     val TEXTURE\_WRAP\_S: Int\n  
 val TEXTURE\_WRAP\_T: Int\n     val TEXTURE\_2D: Int\n     val TEXTURE: Int\n     val  
 TEXTURE\_CUBE\_MAP: Int\n     val TEXTURE\_BINDING\_CUBE\_MAP: Int\n     val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_X: Int\n     val TEXTURE\_CUBE\_MAP\_NEGATIVE\_X: Int\n     val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_Y: Int\n     val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Y: Int\n     val  
 TEXTURE\_CUBE\_MAP\_POSITIVE\_Z: Int\n     val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Z: Int\n     val  
 MAX\_CUBE\_MAP\_TEXTURE\_SIZE: Int\n  
    val TEXTURE0: Int\n     val TEXTURE1: Int\n     val TEXTURE2: Int\n     val TEXTURE3: Int\n  
 val TEXTURE4: Int\n     val TEXTURE5: Int\n     val TEXTURE6: Int\n     val TEXTURE7: Int\n     val  
 TEXTURE8: Int\n     val TEXTURE9: Int\n     val TEXTURE10: Int\n     val TEXTURE11: Int\n     val  
 TEXTURE12: Int\n     val TEXTURE13: Int\n     val TEXTURE14: Int\n     val TEXTURE15: Int\n     val  
 TEXTURE16: Int\n     val TEXTURE17: Int\n     val TEXTURE18: Int\n     val TEXTURE19: Int\n     val  
 TEXTURE20: Int\n     val TEXTURE21: Int\n     val TEXTURE22: Int\n     val TEXTURE23: Int\n     val  
 TEXTURE24: Int\n     val TEXTURE25: Int\n     val TEXTURE26: Int\n     val TEXTURE27: Int\n     val  
 TEXTURE28: Int\n     val TEXTURE29: Int\n     val TEXTURE30: Int\n     val TEXTURE31: Int\n     val  
 ACTIVE\_TEXTURE: Int\n     val REPEAT: Int\n     val CLAMP\_TO\_EDGE: Int\n     val  
 MIRRORED\_REPEAT:  
 Int\n     val FLOAT\_VEC2: Int\n     val FLOAT\_VEC3: Int\n     val FLOAT\_VEC4: Int\n     val  
 INT\_VEC2: Int\n     val INT\_VEC3: Int\n     val INT\_VEC4: Int\n     val BOOL: Int\n     val BOOL\_VEC2:  
 Int\n     val BOOL\_VEC3: Int\n     val BOOL\_VEC4: Int\n     val FLOAT\_MAT2: Int\n     val  
 FLOAT\_MAT3: Int\n     val FLOAT\_MAT4: Int\n     val SAMPLER\_2D: Int\n     val SAMPLER\_CUBE:  
 Int\n     val VERTEX\_ATTRIB\_ARRAY\_ENABLED: Int\n     val VERTEX\_ATTRIB\_ARRAY\_SIZE: Int\n

```

val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val
VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n val VERTEX_ATTRIB_ARRAY_POINTER: Int\n
val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int\n val
IMPLEMENTATION_COLOR_READ_TYPE: Int\n val IMPLEMENTATION_COLOR_READ_FORMAT:
Int\n val COMPILE_STATUS: Int\n val LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val
HIGH_FLOAT: Int\n val LOW_INT: Int\n
 val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val
RENDERBUFFER: Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val
DEPTH_COMPONENT16: Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val
DEPTH_STENCIL: Int\n val RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n
 val RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val
RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val
RENDERBUFFER_ALPHA_SIZE: Int\n val RENDERBUFFER_DEPTH_SIZE: Int\n val
RENDERBUFFER_STENCIL_SIZE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n
val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n val
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n val
FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n val COLOR_ATTACHMENT0:
Int\n val DEPTH_ATTACHMENT: Int\n
 val STENCIL_ATTACHMENT: Int\n val DEPTH_STENCIL_ATTACHMENT: Int\n val NONE: Int\n
 val FRAMEBUFFER_COMPLETE: Int\n val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n
 val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int\n val
FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n val FRAMEBUFFER_UNSUPPORTED: Int\n
val FRAMEBUFFER_BINDING: Int\n val RENDERBUFFER_BINDING: Int\n val
MAX_RENDERBUFFER_SIZE: Int\n val INVALID_FRAMEBUFFER_OPERATION: Int\n val
UNPACK_FLIP_Y_WEBGL: Int\n val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n val
CONTEXT_LOST_WEBGL: Int\n val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n val
BROWSER_DEFAULT_WEBGL: Int\n }\n}\n\n/**\n * Exposes the JavaScript
[WebGLRenderingContext](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext) to Kotlin\n
*\npublic external abstract class WebGLRenderingContext : WebGLRenderingContextBase, RenderingContext {\n
companion object
{\n
 val DEPTH_BUFFER_BIT: Int\n val STENCIL_BUFFER_BIT: Int\n val
COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES: Int\n val LINE_LOOP: Int\n val
LINE_STRIP: Int\n val TRIANGLES: Int\n val TRIANGLE_STRIP: Int\n val TRIANGLE_FAN:
Int\n val ZERO: Int\n val ONE: Int\n val SRC_COLOR: Int\n val ONE_MINUS_SRC_COLOR:
Int\n val SRC_ALPHA: Int\n val ONE_MINUS_SRC_ALPHA: Int\n val DST_ALPHA: Int\n val
ONE_MINUS_DST_ALPHA: Int\n val DST_COLOR: Int\n val ONE_MINUS_DST_COLOR: Int\n
val SRC_ALPHA_SATURATE: Int\n val FUNC_ADD: Int\n val BLEND_EQUATION: Int\n val
BLEND_EQUATION_RGB: Int\n val BLEND_EQUATION_ALPHA: Int\n val FUNC_SUBTRACT:
Int\n val FUNC_REVERSE_SUBTRACT: Int\n val BLEND_DST_RGB: Int\n val
BLEND_SRC_RGB: Int\n val BLEND_DST_ALPHA: Int\n val BLEND_SRC_ALPHA: Int\n
 val CONSTANT_COLOR: Int\n val ONE_MINUS_CONSTANT_COLOR: Int\n val
CONSTANT_ALPHA: Int\n val ONE_MINUS_CONSTANT_ALPHA: Int\n val BLEND_COLOR: Int\n
 val ARRAY_BUFFER: Int\n val ELEMENT_ARRAY_BUFFER: Int\n val
ARRAY_BUFFER_BINDING: Int\n val ELEMENT_ARRAY_BUFFER_BINDING: Int\n val
STREAM_DRAW: Int\n val STATIC_DRAW: Int\n val DYNAMIC_DRAW: Int\n val
BUFFER_SIZE: Int\n val BUFFER_USAGE: Int\n val CURRENT_VERTEX_ATTRIB: Int\n val
FRONT: Int\n val BACK: Int\n val FRONT_AND_BACK: Int\n val CULL_FACE: Int\n val
BLEND: Int\n val DITHER: Int\n val STENCIL_TEST: Int\n val DEPTH_TEST: Int\n val

```

SCISSOR\_TEST: Int\n     val POLYGON\_OFFSET\_FILL: Int\n     val SAMPLE\_ALPHA\_TO\_COVERAGE:  
 Int\n     val SAMPLE\_COVERAGE: Int\n     val NO\_ERROR: Int\n     val INVALID\_ENUM: Int\n     val  
 INVALID\_VALUE: Int\n     val INVALID\_OPERATION:  
 Int\n     val OUT\_OF\_MEMORY: Int\n     val CW: Int\n     val CCW: Int\n     val LINE\_WIDTH: Int\n  
 val ALIASED\_POINT\_SIZE\_RANGE: Int\n     val ALIASED\_LINE\_WIDTH\_RANGE: Int\n     val  
 CULL\_FACE\_MODE: Int\n     val FRONT\_FACE: Int\n     val DEPTH\_RANGE: Int\n     val  
 DEPTH\_WRITEMASK: Int\n     val DEPTH\_CLEAR\_VALUE: Int\n     val DEPTH\_FUNC: Int\n     val  
 STENCIL\_CLEAR\_VALUE: Int\n     val STENCIL\_FUNC: Int\n     val STENCIL\_FAIL: Int\n     val  
 STENCIL\_PASS\_DEPTH\_FAIL: Int\n     val STENCIL\_PASS\_DEPTH\_PASS: Int\n     val STENCIL\_REF:  
 Int\n     val STENCIL\_VALUE\_MASK: Int\n     val STENCIL\_WRITEMASK: Int\n     val  
 STENCIL\_BACK\_FUNC: Int\n     val STENCIL\_BACK\_FAIL: Int\n     val  
 STENCIL\_BACK\_PASS\_DEPTH\_FAIL: Int\n     val STENCIL\_BACK\_PASS\_DEPTH\_PASS: Int\n     val  
 STENCIL\_BACK\_REF: Int\n     val STENCIL\_BACK\_VALUE\_MASK: Int\n     val  
 STENCIL\_BACK\_WRITEMASK: Int\n     val VIEWPORT: Int\n  
    val SCISSOR\_BOX: Int\n     val COLOR\_CLEAR\_VALUE: Int\n     val COLOR\_WRITEMASK: Int\n  
 val UNPACK\_ALIGNMENT: Int\n     val PACK\_ALIGNMENT: Int\n     val MAX\_TEXTURE\_SIZE: Int\n  
 val MAX\_VIEWPORT\_DIMS: Int\n     val SUBPIXEL\_BITS: Int\n     val RED\_BITS: Int\n     val  
 GREEN\_BITS: Int\n     val BLUE\_BITS: Int\n     val ALPHA\_BITS: Int\n     val DEPTH\_BITS: Int\n     val  
 STENCIL\_BITS: Int\n     val POLYGON\_OFFSET\_UNITS: Int\n     val POLYGON\_OFFSET\_FACTOR: Int\n  
    val TEXTURE\_BINDING\_2D: Int\n     val SAMPLE\_BUFFERS: Int\n     val SAMPLES: Int\n     val  
 SAMPLE\_COVERAGE\_VALUE: Int\n     val SAMPLE\_COVERAGE\_INVERT: Int\n     val  
 COMPRESSED\_TEXTURE\_FORMATS: Int\n     val DONT\_CARE: Int\n     val FASTEST: Int\n     val  
 NICEST: Int\n     val GENERATE\_MIPMAP\_HINT: Int\n     val BYTE: Int\n     val UNSIGNED\_BYTE:  
 Int\n     val SHORT: Int\n     val UNSIGNED\_SHORT: Int\n     val INT: Int\n     val  
 UNSIGNED\_INT: Int\n     val FLOAT: Int\n     val DEPTH\_COMPONENT: Int\n     val ALPHA: Int\n  
 val RGB: Int\n     val RGBA: Int\n     val LUMINANCE: Int\n     val LUMINANCE\_ALPHA: Int\n     val  
 UNSIGNED\_SHORT\_4\_4\_4\_4: Int\n     val UNSIGNED\_SHORT\_5\_5\_5\_1: Int\n     val  
 UNSIGNED\_SHORT\_5\_6\_5: Int\n     val FRAGMENT\_SHADER: Int\n     val VERTEX\_SHADER: Int\n  
 val MAX\_VERTEX\_ATTRIBS: Int\n     val MAX\_VERTEX\_UNIFORM\_VECTORS: Int\n     val  
 MAX\_VARYING\_VECTORS: Int\n     val MAX\_COMBINED\_TEXTURE\_IMAGE\_UNITS: Int\n     val  
 MAX\_VERTEX\_TEXTURE\_IMAGE\_UNITS: Int\n     val MAX\_TEXTURE\_IMAGE\_UNITS: Int\n     val  
 MAX\_FRAGMENT\_UNIFORM\_VECTORS: Int\n     val SHADER\_TYPE: Int\n     val DELETE\_STATUS:  
 Int\n     val LINK\_STATUS: Int\n     val VALIDATE\_STATUS: Int\n     val ATTACHED\_SHADERS: Int\n  
    val ACTIVE\_UNIFORMS: Int\n     val ACTIVE\_ATTRIBUTES: Int\n     val  
 SHADING\_LANGUAGE\_VERSION: Int\n     val CURRENT\_PROGRAM:  
 Int\n     val NEVER: Int\n     val LESS: Int\n     val EQUAL: Int\n     val LEQUAL: Int\n     val  
 GREATER: Int\n     val NOTEQUAL: Int\n     val GEQUAL: Int\n     val ALWAYS: Int\n     val KEEP:  
 Int\n     val REPLACE: Int\n     val INCR: Int\n     val DECR: Int\n     val INVERT: Int\n     val  
 INCR\_WRAP: Int\n     val DECR\_WRAP: Int\n     val VENDOR: Int\n     val RENDERER: Int\n     val  
 VERSION: Int\n     val NEAREST: Int\n     val LINEAR: Int\n     val NEAREST\_MIPMAP\_NEAREST: Int\n  
    val LINEAR\_MIPMAP\_NEAREST: Int\n     val NEAREST\_MIPMAP\_LINEAR: Int\n     val  
 LINEAR\_MIPMAP\_LINEAR: Int\n     val TEXTURE\_MAG\_FILTER: Int\n     val TEXTURE\_MIN\_FILTER:  
 Int\n     val TEXTURE\_WRAP\_S: Int\n     val TEXTURE\_WRAP\_T: Int\n     val TEXTURE\_2D: Int\n  
 val TEXTURE: Int\n     val TEXTURE\_CUBE\_MAP: Int\n     val TEXTURE\_BINDING\_CUBE\_MAP: Int\n  
    val TEXTURE\_CUBE\_MAP\_POSITIVE\_X: Int\n  
    val TEXTURE\_CUBE\_MAP\_NEGATIVE\_X: Int\n     val TEXTURE\_CUBE\_MAP\_POSITIVE\_Y: Int\n  
 val TEXTURE\_CUBE\_MAP\_NEGATIVE\_Y: Int\n     val TEXTURE\_CUBE\_MAP\_POSITIVE\_Z: Int\n     val  
 TEXTURE\_CUBE\_MAP\_NEGATIVE\_Z: Int\n     val MAX\_CUBE\_MAP\_TEXTURE\_SIZE: Int\n     val



```

\invisible_member\)\n@kotlin.internal.InlineOnly\npublic inline fun
WebGLContextEventInit(statusMessage: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): WebGLContextEventInit {\n val o = js("{}")\n o["statusMessage"] =
statusMessage\n o["bubbles"]
= bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes
the JavaScript [ArrayBuffer](https://developer.mozilla.org/en/docs/Web/API/ArrayBuffer) to Kotlin\n */\npublic
external open class ArrayBuffer(length: Int) : BufferDataSource {\n open val byteLength: Int\n fun slice(begin:
Int, end: Int = definedExternally): ArrayBuffer\n\n companion object {\n fun isView(value: Any?): Boolean\n
 }\n}\n\n/**\n * Exposes the JavaScript
[ArrayBufferView](https://developer.mozilla.org/en/docs/Web/API/ArrayBufferView) to Kotlin\n */\npublic
external interface ArrayBufferView : BufferDataSource {\n val buffer: ArrayBuffer\n val byteOffset: Int\n val
byteLength: Int\n}\n\n/**\n * Exposes the JavaScript
[Int8Array](https://developer.mozilla.org/en/docs/Web/API/Int8Array) to Kotlin\n */\npublic external open class
Int8Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int8Array)\n
 constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Int8Array, offset: Int = definedExternally)\n fun set(array:
Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int8Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin\n
*/\npublic external open class Uint8Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array:
Uint8Array)\n constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n
 override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint8Array, offset: Int =
definedExternally)\n fun set(array: Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end:
Int): Uint8Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.set(index: Int,
value: Byte) { asDynamic()[index]
= value }\n\n/**\n * Exposes the JavaScript
[Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin\n */\npublic
external open class Uint8ClampedArray : ArrayBufferView {\n constructor(length: Int)\n constructor(array:
Uint8ClampedArray)\n constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n
 override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint8ClampedArray, offset: Int =
definedExternally)\n fun set(array: Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end:
Int): Uint8ClampedArray\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]\n\n@Suppress("invisible_reference",
"invisible_member")\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript

```

```

[Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin\n *\npublic external open class
Int16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int16Array)\n
constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Int16Array, offset: Int = definedExternally)\n fun set(array:
Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int16Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin\n *\npublic external open
class Uint16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint16Array)\n
constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Uint16Array, offset: Int = definedExternally)\n fun set(array:
Array<Short>, offset: Int = definedExternally)\n
fun subarray(start: Int, end: Int): Uint16Array\n\n companion object {\n val BYTES_PER_ELEMENT:
Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin\n *\npublic external open class
Int32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int32Array)\n
constructor(array: Array<Int>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n
override
val byteLength: Int\n fun set(array: Int32Array, offset: Int = definedExternally)\n fun set(array: Array<Int>,
offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int32Array\n\n companion object {\n val
BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.get(index: Int): Int
= asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin\n *\npublic external open
class Uint32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint32Array)\n
constructor(array: Array<Int>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n
override val byteLength: Int\n fun set(array: Uint32Array, offset: Int = definedExternally)\n fun set(array:
Array<Int>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint32Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.get(index: Int):
Int = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin\n *\npublic external open
class Float32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array:

```

```

Float32Array)\n constructor(array: Array<Float>)\n constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n
override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Float32Array, offset: Int =
definedExternally)\n fun set(array: Array<Float>, offset: Int = definedExternally)\n fun subarray(start: Int, end:
Int): Float32Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.get(index: Int):
Float = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.set(index: Int,
value: Float) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array)
to Kotlin\n */\npublic external open class Float64Array : ArrayBufferView {\n constructor(length: Int)\n
constructor(array: Float64Array)\n constructor(array: Array<Double>)\n constructor(buffer: ArrayBuffer,
byteOffset: Int = definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer:
ArrayBuffer\n override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Float64Array, offset:
Int = definedExternally)\n fun set(array: Array<Double>, offset: Int = definedExternally)\n fun subarray(start:
Int, end: Int): Float64Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float64Array.get(index: Int):
Double = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator
fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin\n */\npublic external open class
DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally) :
ArrayBufferView {\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n override val
byteLength: Int\n fun getInt8(byteOffset: Int): Byte\n fun getUint8(byteOffset: Int): Byte\n fun
getInt16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short\n fun getUint16(byteOffset: Int,
littleEndian: Boolean = definedExternally): Short\n fun getInt32(byteOffset: Int, littleEndian: Boolean =
definedExternally): Int\n fun getUint32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int\n fun
getFloat32(byteOffset: Int, littleEndian: Boolean = definedExternally): Float\n fun getFloat64(byteOffset: Int,
littleEndian: Boolean = definedExternally): Double\n fun setInt8(byteOffset: Int, value: Byte)\n fun
setUint8(byteOffset: Int, value: Byte)\n fun setInt16(byteOffset: Int, value: Short, littleEndian: Boolean =
definedExternally)\n fun setUint16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)\n
fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)\n fun setUint32(byteOffset: Int,
value: Int, littleEndian: Boolean = definedExternally)\n fun setFloat32(byteOffset: Int, value: Float, littleEndian:
Boolean = definedExternally)\n fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean =
definedExternally)\n}\n\npublic external interface BufferDataSource\n\npublic external interface
TexImageSource", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt
file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for
details\n\npackage org.w3c.dom.clipboard\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface ClipboardEventInit : EventInit {\n var
clipboardData: DataTransfer? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ClipboardEventInit(clipboardData:
DataTransfer? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
ClipboardEventInit {\n val o = js("{}")\n o["clipboardData"] = clipboardData\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript

```



```

[ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin\n */\npublic external open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit =
definedExternally) : Event {\n open val clipboardData: DataTransfer?\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to Kotlin\n */\npublic external abstract class
Clipboard : EventTarget {\n fun read(): Promise<DataTransfer>\n fun readText(): Promise<String>\n fun
write(data: DataTransfer): Promise<Unit>\n fun writeText(data: String): Promise<Unit>\n}\n\n\npublic external
interface ClipboardPermissionDescriptor {\n var allowWithoutGesture: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
ClipboardPermissionDescriptor(allowWithoutGesture:
Boolean? = false): ClipboardPermissionDescriptor {\n val o = js(\"({})\")\n o[\"allowWithoutGesture\"] =
allowWithoutGesture\n return o\n}\", \"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See
github.com/kotlin/dukat for details\n\npackage org.w3c.dom.css\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\n\npublic external abstract class MediaList : ItemArrayLike<String>
{\n open var mediaText: String\n fun appendMedium(medium: String)\n fun deleteMedium(medium: String)\n
 override fun item(index: Int): String?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MediaList.get(index: Int):
String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin\n
*/\npublic external abstract class StyleSheet {\n open val type: String\n open val href: String?\n open val
ownerNode: UnionElementOrProcessingInstruction?\n open val parentStyleSheet: StyleSheet?\n open val title:
String?\n open val media: MediaList\n open var disabled: Boolean\n}\n\n\n/**\n * Exposes the JavaScript
[CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet) to Kotlin\n */\npublic external
abstract class CSSStyleSheet : StyleSheet {\n open val ownerRule: CSSRule?\n open val cssRules:
CSSRuleList\n fun insertRule(rule: String, index: Int): Int\n fun deleteRule(index: Int)\n}\n\n\n/**\n * Exposes the
JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin\n */\npublic
external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {\n override fun item(index:
Int): StyleSheet?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun StyleSheetList.get(index: Int):
StyleSheet? = asDynamic()[index]\n\n\n/**\n * Exposes the JavaScript
[LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin\n */\npublic external interface
LinkStyle {\n val sheet: StyleSheet?\n get() = definedExternally\n}\n\n\n/**\n * Exposes the JavaScript
[CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin\n */\npublic external abstract
class CSSRuleList : ItemArrayLike<CSSRule> {\n override fun item(index: Int):
CSSRule?\n}\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun CSSRuleList.get(index: Int):
CSSRule? = asDynamic()[index]\n\n\n/**\n * Exposes the JavaScript
[CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin\n */\npublic external abstract class
CSSRule
{\n open val type: Short\n open var cssText: String\n open val parentRule: CSSRule?\n open val
parentStyleSheet: CSSStyleSheet?\n\n companion object {\n val STYLE_RULE: Short\n val
CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val
FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val
NAMESPACE_RULE: Short\n }\n}\n\n\n/**\n * Exposes the JavaScript

```

[CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin\n \*^public external abstract class CSSStyleRule : CSSRule {\n open var selectorText: String\n open val style: CSSStyleDeclaration\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\npublic external abstract class CSSImportRule : CSSRule {\n open val href: String\n open val media: MediaList\n open val styleSheet: CSSStyleSheet\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript

[CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin\n \*^public external abstract class CSSGroupingRule : CSSRule {\n open val cssRules: CSSRuleList\n fun insertRule(rule: String, index: Int): Int\n fun deleteRule(index: Int)\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript

[CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin\n \*^public external abstract class CSSMediaRule : CSSGroupingRule {\n open val media: MediaList\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript

[CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin\n \*^public external abstract class CSSPageRule : CSSGroupingRule {\n open var selectorText: String\n open val style: CSSStyleDeclaration\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\npublic external abstract class CSSMarginRule : CSSRule {\n open val name: String\n open val style: CSSStyleDeclaration\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript

[CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin\n \*^public external abstract class CSSNamespaceRule : CSSRule {\n open val namespaceURI: String\n open val prefix: String\n\n companion object {\n val STYLE\_RULE: Short\n val CHARSET\_RULE: Short\n val IMPORT\_RULE: Short\n val MEDIA\_RULE: Short\n val FONT\_FACE\_RULE: Short\n val PAGE\_RULE: Short\n val MARGIN\_RULE: Short\n val NAMESPACE\_RULE: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript

[CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin\n \*^public external abstract class CSSStyleDeclaration : ItemArrayLike<String> {\n open var cssText: String\n open val parentRule: CSSRule?\n open var cssFloat: String\n open var alignContent: String\n open var alignItems: String\n open var alignSelf: String\n open var animation: String\n open var animationDelay: String\n open var animationDirection: String\n open var animationDuration: String\n open var animationFillMode: String\n open var animationIterationCount: String\n open var animationName: String\n open var animationPlayState: String\n open var animationTimingFunction: String\n open var backfaceVisibility: String\n open var background: String\n open var backgroundAttachment: String\n open var backgroundClip: String\n open var backgroundColor: String\n open var backgroundImage: String\n open var backgroundOrigin: String\n open var backgroundPosition: String\n open var backgroundRepeat: String\n open

var backgroundSize: String\n open var border: String\n open var borderBottom: String\n open var  
borderBottomColor: String\n open var borderBottomLeftRadius: String\n open var borderBottomRightRadius:  
String\n open var borderBottomStyle: String\n open var borderBottomWidth: String\n open var  
borderCollapse: String\n open var borderColor: String\n open var borderImage: String\n open var  
borderImageOutset: String\n open var borderImageRepeat: String\n open var borderImageSlice: String\n open  
var borderImageSource: String\n open var borderImageWidth: String\n open var borderLeft: String\n open var  
borderLeftColor: String\n open var borderLeftStyle: String\n open var borderLeftWidth: String\n open var  
borderRadius:  
String\n open var borderRight: String\n open var borderRightColor: String\n open var borderRightStyle:  
String\n open var borderRightWidth: String\n open var borderSpacing: String\n open var borderStyle: String\n  
open var borderTop: String\n open var borderTopColor: String\n open var borderTopLeftRadius: String\n  
open var borderTopRightRadius: String\n open var borderTopStyle: String\n open var borderTopWidth: String\n  
open var borderWidth: String\n open var bottom: String\n open var boxDecorationBreak: String\n open var  
boxShadow: String\n open var boxSizing: String\n open var breakAfter: String\n open var breakBefore:  
String\n open var breakInside: String\n open var captionSide: String\n open var clear: String\n open var clip:  
String\n open var color: String\n open var columnCount: String\n open var columnFill: String\n open var  
columnGap: String\n open var columnRule: String\n open var columnRuleColor:  
String\n open var columnRuleStyle: String\n open var columnRuleWidth: String\n open var columnSpan:  
String\n open var columnWidth: String\n open var columns: String\n open var content: String\n open var  
counterIncrement: String\n open var counterReset: String\n open var cursor: String\n open var direction:  
String\n open var display: String\n open var emptyCells: String\n open var filter: String\n open var flex:  
String\n open var flexBasis: String\n open var flexDirection: String\n open var flexFlow: String\n open var  
flexGrow: String\n open var flexShrink: String\n open var flexWrap: String\n open var font: String\n open  
var fontFamily: String\n open var fontFeatureSettings: String\n open var fontKerning: String\n open var  
fontLanguageOverride: String\n open var fontSize: String\n open var fontSizeAdjust: String\n open var  
fontStretch: String\n open var fontStyle: String\n open var fontSynthesis:  
String\n open var fontVariant: String\n open var fontVariantAlternates: String\n open var fontVariantCaps:  
String\n open var fontVariantEastAsian: String\n open var fontVariantLigatures: String\n open var  
fontVariantNumeric: String\n open var fontVariantPosition: String\n open var fontWeight: String\n open var  
hangingPunctuation: String\n open var height: String\n open var hyphens: String\n open var imageOrientation:  
String\n open var imageRendering: String\n open var imageResolution: String\n open var imeMode: String\n  
open var justifyContent: String\n open var left: String\n open var letterSpacing: String\n open var lineBreak:  
String\n open var lineHeight: String\n open var listStyle: String\n open var listStyleImage: String\n open var  
listStylePosition: String\n open var listStyleType: String\n open var margin: String\n open var marginBottom:  
String\n open var marginLeft: String\n open var marginRight:  
String\n open var marginTop: String\n open var mark: String\n open var markAfter: String\n open var  
markBefore: String\n open var marks: String\n open var marqueeDirection: String\n open var  
marqueePlayCount: String\n open var marqueeSpeed: String\n open var marqueeStyle: String\n open var  
mask: String\n open var maskType: String\n open var maxHeight: String\n open var maxWidth: String\n  
open var minHeight: String\n open var minWidth: String\n open var navDown: String\n open var navIndex:  
String\n open var navLeft: String\n open var navRight: String\n open var navUp: String\n open var objectFit:  
String\n open var objectPosition: String\n open var opacity: String\n open var order: String\n open var  
orphans: String\n open var outline: String\n open var outlineColor: String\n open var outlineOffset: String\n  
open var outlineStyle: String\n open var outlineWidth: String\n open var overflowWrap:  
String\n open var overflowX: String\n open var overflowY: String\n open var padding: String\n open var  
paddingBottom: String\n open var paddingLeft: String\n open var paddingRight: String\n open var  
paddingTop: String\n open var pageBreakAfter: String\n open var pageBreakBefore: String\n open var  
pageBreakInside: String\n open var perspective: String\n open var perspectiveOrigin: String\n open var

```

phonemes: String\n open var position: String\n open var quotes: String\n open var resize: String\n open var
rest: String\n open var restAfter: String\n open var restBefore: String\n open var right: String\n open var
tabSize: String\n open var tableLayout: String\n open var textAlign: String\n open var textAlignLast: String\n
open var textCombineUpright: String\n open var textDecoration: String\n open var textDecorationColor:
String\n open var textDecorationLine: String\n open var textDecorationStyle:
String\n open var textIndent: String\n open var textJustify: String\n open var textOrientation: String\n open
var textOverflow: String\n open var textShadow: String\n open var textTransform: String\n open var
textUnderlinePosition: String\n open var top: String\n open var transform: String\n open var transformOrigin:
String\n open var transformStyle: String\n open var transition: String\n open var transitionDelay: String\n
open var transitionDuration: String\n open var transitionProperty: String\n open var transitionTimingFunction:
String\n open var unicodeBidi: String\n open var verticalAlign: String\n open var visibility: String\n open
var voiceBalance: String\n open var voiceDuration: String\n open var voicePitch: String\n open var
voicePitchRange: String\n open var voiceRate: String\n open var voiceStress: String\n open var voiceVolume:
String\n open var whiteSpace: String\n open var widows: String\n
open var width: String\n open var wordBreak: String\n open var wordSpacing: String\n open var wordWrap:
String\n open var writingMode: String\n open var zIndex: String\n open var _dashed_attribute: String\n open
var _camel_cased_attribute: String\n open var _webkit_cased_attribute: String\n fun getPropertyValue(property:
String): String\n fun getPropertyPriority(property: String): String\n fun setProperty(property: String, value:
String, priority: String = definedExternally)\n fun setPropertyValue(property: String, value: String)\n fun
setPropertyPriority(property: String, priority: String)\n fun removeProperty(property: String): String\n override
fun item(index: Int): String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]\n\npublic external interface
ElementCSSInlineStyle {\n val style: CSSStyleDeclaration\n}\n\n/**\n
* Exposes the JavaScript [CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin\n */\npublic
external abstract class CSS {\n companion object {\n fun escape(ident: String): String\n }\n}\n\npublic
external interface UnionElementOrProcessingInstruction\", \"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.encryptedmedia\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript
[MediaKeySystemConfiguration](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemConfiguration)
to Kotlin\n */\n\npublic external interface MediaKeySystemConfiguration {\n var label: String? /* = \"\" *\n
get() = definedExternally\n set(value) = definedExternally\n var initDataTypes: Array<String>? /* =
arrayOf() *\n get() = definedExternally\n set(value) = definedExternally\n var audioCapabilities:
Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n get() = definedExternally\n set(value) =
definedExternally\n var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n
get() = definedExternally\n set(value) = definedExternally\n var distinctiveIdentifier:
MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL *\n get() = definedExternally\n
set(value) = definedExternally\n var persistentState: MediaKeysRequirement? /* =
MediaKeysRequirement.OPTIONAL *\n get() = definedExternally\n set(value) = definedExternally\n
var sessionTypes: Array<String>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaKeySystemConfiguration(label:
String? = \"\", initDataTypes: Array<String>? = arrayOf(), audioCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), videoCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), distinctiveIdentifier: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, persistentState: MediaKeysRequirement? =

```

```

MediaKeysRequirement.OPTIONAL, sessionTypes: Array<String>? = undefined): MediaKeySystemConfiguration
{\n val o = js(\("{ }\")\n o["label"] = label\n o["initDataTypes"] = initDataTypes\n
o["audioCapabilities"] = audioCapabilities\n o["videoCapabilities"] = videoCapabilities\n
o["distinctiveIdentifier"] = distinctiveIdentifier\n o["persistentState"] = persistentState\n o["sessionTypes"]
= sessionTypes\n return o\n}\n\npublic external interface MediaKeySystemMediaCapability {\n var
contentType: String? /* = \"" */\n
get() = definedExternally\n set(value) = definedExternally\n var robustness: String? /* = \"" */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaKeySystemMediaCapability(contentType: String? = \\"", robustness: String? = \"):
MediaKeySystemMediaCapability {\n val o = js(\("{ }\")\n o["contentType"] = contentType\n
o["robustness"] = robustness\n return o\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySystemAccess](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemAccess) to Kotlin\n
*/\n\npublic external abstract class MediaKeySystemAccess {\n open val keySystem: String\n fun
getConfiguration(): MediaKeySystemConfiguration\n fun createMediaKeys(): Promise<MediaKeys>\n}\n\n/**\n
* Exposes the JavaScript [MediaKeys](https://developer.mozilla.org/en/docs/Web/API/MediaKeys) to Kotlin\n
*/\n\npublic external
abstract class MediaKeys {\n fun createSession(sessionType: MediaKeySessionType = definedExternally):
MediaKeySession\n fun setServerCertificate(serverCertificate: dynamic): Promise<Boolean>\n}\n\n/**\n *
Exposes the JavaScript [MediaKeySession](https://developer.mozilla.org/en/docs/Web/API/MediaKeySession) to
Kotlin\n */\n\npublic external abstract class MediaKeySession : EventTarget {\n open val sessionId: String\n open
val expiration: Double\n open val closed: Promise<Unit>\n open val keyStatuses: MediaKeyStatusMap\n open
var onkeystatuschange: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun
generateRequest(initDataType: String, initData: dynamic): Promise<Unit>\n fun load(sessionId: String):
Promise<Boolean>\n fun update(response: dynamic): Promise<Unit>\n fun close(): Promise<Unit>\n fun
remove(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyStatusMap](https://developer.mozilla.org/en/docs/Web/API/MediaKeyStatusMap)
to Kotlin\n */\n\npublic external abstract class MediaKeyStatusMap {\n open val size: Int\n fun has(keyId:
dynamic): Boolean\n fun get(keyId: dynamic): Any?\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyMessageEvent](https://developer.mozilla.org/en/docs/Web/API/MediaKeyMessageEvent) to Kotlin\n
*/\n\npublic external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) :
Event {\n open val messageType: MediaKeyMessageType\n open val message: ArrayBuffer\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaKeyMessageEventInit : EventInit {\n var
messageType: MediaKeyMessageType?\n var message:
ArrayBuffer?\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaKeyMessageEventInit(messageType: MediaKeyMessageType?, message:
ArrayBuffer?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MediaKeyMessageEventInit {\n val o = js(\("{ }\")\n o["messageType"] = messageType\n o["message"] =
message\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\npublic external open class MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit =
definedExternally) : Event {\n open val initDataType: String\n open val initData: ArrayBuffer?\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaEncryptedEventInit : EventInit {\n var
initDataType: String? /* = \"" */\n get() = definedExternally\n set(value) = definedExternally\n var
initData: ArrayBuffer? /* = null */\n get() = definedExternally\n set(value) =

```

```

definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaEncryptedEventInit(initDataType: String? = "", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): MediaEncryptedEventInit {\n val o = js("{}")\n
o["initDataType"] = initDataType\n o["initData"] = initData\n o["bubbles"] = bubbles\n o["cancelable"]
= cancelable\n o["composed"] = composed\n return o\n}\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaKeysRequirement {\n companion object\n}\n\npublic inline val
MediaKeysRequirement.Companion.REQUIRED: MediaKeysRequirement get() =
"required".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.OPTIONAL: MediaKeysRequirement get() =
"optional".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic
inline val MediaKeysRequirement.Companion.NOT_ALLOWED: MediaKeysRequirement get() = "not-
allowed".asDynamic().unsafeCast<MediaKeysRequirement>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaKeySessionType {\n companion object\n}\n\npublic inline val
MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() =
"temporary".asDynamic().unsafeCast<MediaKeySessionType>()\n\npublic inline val
MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = "persistent-
license".asDynamic().unsafeCast<MediaKeySessionType>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaKeyStatus {\n companion object\n}\n\npublic inline val MediaKeyStatus.Companion.USABLE:
MediaKeyStatus get() = "usable".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic
inline val MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() =
"expired".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() =
"released".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = "output-
restricted".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = "output-
downscaled".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = "status-
pending".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = "internal-
error".asDynamic().unsafeCast<MediaKeyStatus>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic
external interface MediaKeyMessageType {\n companion object\n}\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = "license-
request".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = "license-
renewal".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType get() = "license-
release".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.INDIVIDUALIZATION_REQUEST: MediaKeyMessageType get() =
"individualization-request".asDynamic().unsafeCast<MediaKeyMessageType>()", /*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n//

```

```

See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.events\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\n\n/**\n * Exposes the JavaScript
[UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin\n *\npublic external open class
UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {\n open val view: Window?\n
open val detail: Int\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n
val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface UIEventInit :
EventInit {\n var view: Window? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n var detail: Int? /* = 0 */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun UIEventInit(view: Window? =
null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
UIEventInit {\n val o = js(\"({})\")\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin\n *\npublic external open class
FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {\n open val relatedTarget:
EventTarget?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface FocusEventInit :
UIEventInit {\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic
inline fun FocusEventInit(relatedTarget: EventTarget? = null, view: Window? = null, detail: Int? = 0, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): FocusEventInit {\n val o =
js(\"({})\")\n o[\"relatedTarget\"] = relatedTarget\n o[\"view\"] = view\n o[\"detail\"] = detail\n
o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript [MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent)
to Kotlin\n *\npublic external open class MouseEvent(type: String, eventInitDict: MouseEventInit =
definedExternally) : UIEvent, UnionElementOrMouseEvent {\n open val screenX: Int\n open val screenY: Int\n
open val clientX: Int\n open val clientY: Int\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n
open val altKey: Boolean\n open val metaKey: Boolean\n open val button: Short\n
open val buttons: Short\n open val relatedTarget: EventTarget?\n open val region: String?\n open val pageX:
Double\n open val pageY: Double\n open val x: Double\n open val y: Double\n open val offsetX: Double\n
open val offsetY: Double\n fun getModifierState(keyArg: String): Boolean\n\n companion object {\n val
NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MouseEventInit : EventModifierInit {\n var
screenX: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var screenY: Int? /*
= 0 */\n get() = definedExternally\n set(value) = definedExternally\n var clientX: Int? /* = 0 */\n
get() = definedExternally\n set(value) = definedExternally\n var clientY: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var button: Short?
/* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var buttons: Short? /* = 0 */\n
get() = definedExternally\n set(value) = definedExternally\n var relatedTarget: EventTarget? /* = null */\n
get() = definedExternally\n set(value) = definedExternally\n var region: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun MouseEventInit(screenX: Int? = 0,
screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget:
EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean?
= false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false,
modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper:

```

```

Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper:
Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? =
null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MouseEventInit {\n val o = js\("\{\}\")\n o["screenX"] = screenX\n o["screenY"] = screenY\n
o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n o["buttons"] = buttons\n
o["relatedTarget"] = relatedTarget\n o["region"] = region\n o["ctrlKey"] = ctrlKey\n o["shiftKey"] =
shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"] =
modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] = modifierHyper\n o["modifierNumLock"] =
modifierNumLock\n
o["modifierScrollLock"] = modifierScrollLock\n o["modifierSuper"] = modifierSuper\n
o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] =
view\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] =
composed\n return o\n}\n\npublic external interface EventModifierInit : UIEventInit {\n var ctrlKey: Boolean?
/* = false */\n get() = definedExternally\n set(value) = definedExternally\n var shiftKey: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var altKey: Boolean? /* = false
/\n get() = definedExternally\n set(value) = definedExternally\n var metaKey: Boolean? / = false */\n
get() = definedExternally\n set(value) = definedExternally\n var modifierAltGraph: Boolean? /* = false */\n
get() = definedExternally\n set(value) = definedExternally\n
var modifierCapsLock: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var modifierFn: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var modifierFnLock: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var modifierHyper: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var modifierNumLock: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierScrollLock: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierSuper: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var modifierSymbol: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n var
modifierSymbolLock: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress\("INVISIBLE_REFERENCE",
\("INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventModifierInit(ctrlKey: Boolean? =
false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? =
false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): EventModifierInit {\n val o = js\("\{\}\")\n o["ctrlKey"] = ctrlKey\n o["shiftKey"] = shiftKey\n
o["altKey"] = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"]
= modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] = modifierHyper\n o["modifierNumLock"] =
modifierNumLock\n o["modifierScrollLock"] = modifierScrollLock\n o["modifierSuper"] = modifierSuper\n
o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] =
view\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] =
composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin\n */\npublic external open
class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {\n open val
deltaX: Double\n open val deltaY: Double\n open val deltaZ: Double\n open val deltaMode: Int\n\ncompanion object {\n

```



```

 val DOM_DELTA_PIXEL: Int\n val DOM_DELTA_LINE: Int\n val DOM_DELTA_PAGE: Int\n
 val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
 BUBBLING_PHASE: Short\n } \n} \n\npublic external interface WheelEventInit : MouseEventInit { \n var
 deltaX: Double? /* = 0.0 */ \n get() = definedExternally \n set(value) = definedExternally \n var deltaY:
 Double? /* = 0.0 */ \n get() = definedExternally \n set(value) = definedExternally \n var deltaZ: Double? /*
 = 0.0 */ \n get() = definedExternally \n set(value) = definedExternally \n var deltaMode: Int? /* = 0 */ \n
 get() = definedExternally \n set(value) = definedExternally \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",
 \"INVISIBLE_MEMBER\") \n@kotlin.internal.InlineOnly \npublic inline fun WheelEventInit(deltaX: Double? = 0.0,
 deltaY: Double? = 0.0, deltaZ: Double? = 0.0, deltaMode: Int? = 0, screenX: Int? = 0, screenY: Int? = 0, clientX:
 Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region:
 String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean?
 = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false,
 modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false,
 modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false,
 modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false,
 cancelable: Boolean? = false, composed: Boolean? = false): WheelEventInit { \n val o = js(\"({})\") \n
 o[\"deltaX\"] = deltaX \n o[\"deltaY\"] = deltaY \n o[\"deltaZ\"] = deltaZ \n o[\"deltaMode\"] = deltaMode \n
 o[\"screenX\"] = screenX \n o[\"screenY\"] = screenY \n o[\"clientX\"] = clientX \n o[\"clientY\"] = clientY \n
 o[\"button\"] = button \n o[\"buttons\"] = buttons \n o[\"relatedTarget\"] = relatedTarget \n o[\"region\"] =
 region \n o[\"ctrlKey\"] = ctrlKey \n o[\"shiftKey\"] = shiftKey \n o[\"altKey\"] = altKey \n o[\"metaKey\"] =
 metaKey \n o[\"modifierAltGraph\"] = modifierAltGraph \n o[\"modifierCapsLock\"] = modifierCapsLock \n
 o[\"modifierFn\"] = modifierFn \n o[\"modifierFnLock\"] = modifierFnLock \n o[\"modifierHyper\"] =
 modifierHyper \n o[\"modifierNumLock\"] = modifierNumLock \n o[\"modifierScrollLock\"] =
 modifierScrollLock \n o[\"modifierSuper\"] = modifierSuper \n o[\"modifierSymbol\"] = modifierSymbol \n
 o[\"modifierSymbolLock\"] = modifierSymbolLock \n o[\"view\"] = view \n o[\"detail\"] = detail \n
 o[\"bubbles\"] = bubbles \n o[\"cancelable\"] = cancelable \n o[\"composed\"] = composed \n return
 o \n} \n\n/** \n * Exposes the JavaScript [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to
 Kotlin \n */ \npublic external
 open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent { \n open val
 data: String \n open val isComposing: Boolean \n\n companion object { \n val NONE: Short \n val
 CAPTURING_PHASE: Short \n val AT_TARGET: Short \n val BUBBLING_PHASE: Short \n
 } \n} \n\npublic external interface InputEventInit : UIEventInit { \n var data: String? /* = \"\" */ \n get() =
 definedExternally \n set(value) = definedExternally \n var isComposing: Boolean? /* = false */ \n get() =
 definedExternally \n set(value) = definedExternally \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",
 \"INVISIBLE_MEMBER\") \n@kotlin.internal.InlineOnly \npublic inline fun InputEventInit(data: String? = \"\",
 isComposing: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable:
 Boolean? = false, composed: Boolean? = false): InputEventInit { \n val o = js(\"({})\") \n o[\"data\"] = data \n
 o[\"isComposing\"]
 = isComposing \n o[\"view\"] = view \n o[\"detail\"] = detail \n o[\"bubbles\"] = bubbles \n o[\"cancelable\"] =
 cancelable \n o[\"composed\"] = composed \n return o \n} \n\n/** \n * Exposes the JavaScript
 [KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin \n */ \npublic external
 open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent { \n
 open val key: String \n open val code: String \n open val location: Int \n open val ctrlKey: Boolean \n open val
 shiftKey: Boolean \n open val altKey: Boolean \n open val metaKey: Boolean \n open val repeat: Boolean \n
 open val isComposing: Boolean \n open val charCode: Int \n open val keyCode: Int \n open val which: Int \n
 fun getModifierState(keyArg: String): Boolean \n\n companion object { \n val
 DOM_KEY_LOCATION_STANDARD: Int \n val DOM_KEY_LOCATION_LEFT: Int \n val
 DOM_KEY_LOCATION_RIGHT: Int \n

```



```

definedExternally)\n fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally)\n
fun removeEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)\n fun
removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)\n fun
dispatchEvent(event: Event): Boolean\n}\n\n**\n * Exposes the JavaScript
[EventListener](https://developer.mozilla.org/en/docs/Web/API/EventListener) to Kotlin\n *\n\npublic external
interface EventListener {\n fun handleEvent(event: Event)\n},"/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.clipboard.*\nimport org.w3c.dom.css.*\nimport
org.w3c.dom.encryptedmedia.*\nimport org.w3c.dom.events.*\nimport org.w3c.dom.mediacapture.*\nimport
org.w3c.dom.mediasource.*\nimport org.w3c.dom.pointerevents.*\nimport org.w3c.dom.svg.*\nimport
org.w3c.fetch.*\nimport org.w3c.files.*\nimport org.w3c.performance.*\nimport org.w3c.workers.*\nimport
org.w3c.xhr.*\n\npublic external abstract class HTMLAllCollection {\n open val length: Int\n fun
item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?\n fun namedItem(name:
String): UnionElementOrHTMLCollection?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? = asDynamic()[name]\n\n**\n *
Exposes the JavaScript
[HTMLFormControlsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLFormControlsCollection)
to
Kotlin\n *\n\npublic external abstract class HTMLFormControlsCollection : HTMLCollection\n\n**\n * Exposes
the JavaScript [RadioNodeList](https://developer.mozilla.org/en/docs/Web/API/RadioNodeList) to Kotlin\n
*\n\npublic external abstract class RadioNodeList : NodeList, UnionElementOrRadioNodeList {\n open var value:
String\n}\n\n**\n * Exposes the JavaScript
[HTMLOptionsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionsCollection) to Kotlin\n
*\n\npublic external abstract class HTMLOptionsCollection : HTMLCollection {\n override var length: Int\n open
var selectedIndex: Int\n fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before:
dynamic = definedExternally)\n fun remove(index: Int)\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n**\n *
Exposes the JavaScript [HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement)
to Kotlin\n *\n\npublic external abstract class HTMLElement : Element, GlobalEventHandlers,
DocumentAndElementEventHandlers, ElementContentEditable, ElementCSSInlineStyle {\n open var title:
String\n open var lang: String\n open var translate: Boolean\n open var dir: String\n open val dataset:
DOMStringMap\n open var hidden: Boolean\n open var tabIndex: Int\n open var accessKey: String\n open
val accessKeyLabel: String\n open var draggable: Boolean\n open val dropzone: DOMTokenList\n open var
contextMenu: HTMLMenuElement?\n open var spellcheck: Boolean\n open var innerText: String\n open val
offsetParent: Element?\n open val offsetTop: Int\n open val offsetLeft: Int\n open val offsetWidth: Int\n open
val offsetHeight: Int\n fun click()\n fun focus()\n fun blur()\n fun forceSpellCheck()\n\n companion object
{\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val

```

```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to Kotlin\n
*\npublic external abstract class HTMLUnknownElement : HTMLElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin\n
*\npublic external
abstract class DOMStringMap\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMStringMap.get(name:
String): String? = asDynamic()[name]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMStringMap.set(name:
String, value: String) { asDynamic()[name] = value }\n\n/**\n * Exposes the JavaScript
[HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin\n
*\npublic
external abstract class HTMLHtmlElement : HTMLElement {\n open var version: String\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to
Kotlin\n
*\npublic external abstract class HTMLHeadElement : HTMLElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin\n
*\npublic external abstract class HTMLTitleElement : HTMLElement {\n open var text: String\n\n companion

```

```

object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/** Exposes the JavaScript [HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement) to Kotlin
 *
 * public external abstract class HTMLElement : HTMLBaseElement {
 * open var href: String
 * open var target: String
 * companion object {
 * val ELEMENT_NODE: Short
 * val ATTRIBUTE_NODE: Short
 * val TEXT_NODE: Short
 * val CDATA_SECTION_NODE: Short
 * val ENTITY_REFERENCE_NODE: Short
 * val ENTITY_NODE: Short
 * val PROCESSING_INSTRUCTION_NODE: Short
 * val COMMENT_NODE: Short
 * val DOCUMENT_NODE: Short
 * val DOCUMENT_TYPE_NODE: Short
 * val DOCUMENT_FRAGMENT_NODE: Short
 * val NOTATION_NODE: Short
 * val DOCUMENT_POSITION_DISCONNECTED: Short
 * val DOCUMENT_POSITION_PRECEDING: Short
 * val DOCUMENT_POSITION_FOLLOWING: Short
 * val DOCUMENT_POSITION_CONTAINS: Short
 * val DOCUMENT_POSITION_CONTAINED_BY: Short
 * val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 * }
 * }

/** Exposes the JavaScript [HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin
 *
 * public external abstract class HTMLLinkElement : HTMLElement, LinkStyle {
 * open var href: String
 * open var crossOrigin: String?
 * open var rel: String
 * open var `as`: RequestDestination
 * open val relList: DOMTokenList
 * open var media: String
 * open var nonce: String
 * open var hreflang: String
 * open var type: String
 * open val sizes: DOMTokenList
 * open varreferrerPolicy: String
 * open var charset: String
 * open var rev: String
 * open var target: String
 * open var scope: String
 * open var workerType: WorkerType
 * companion object {
 * val ELEMENT_NODE: Short
 * val ATTRIBUTE_NODE: Short
 * val TEXT_NODE: Short
 * val CDATA_SECTION_NODE: Short
 * val ENTITY_REFERENCE_NODE: Short
 * val ENTITY_NODE: Short
 * val PROCESSING_INSTRUCTION_NODE: Short
 * val COMMENT_NODE: Short
 * val DOCUMENT_NODE: Short
 * val DOCUMENT_TYPE_NODE: Short
 * val DOCUMENT_FRAGMENT_NODE: Short
 * val NOTATION_NODE: Short
 * val DOCUMENT_POSITION_DISCONNECTED: Short
 * val DOCUMENT_POSITION_PRECEDING: Short
 * val DOCUMENT_POSITION_FOLLOWING: Short
 * val DOCUMENT_POSITION_CONTAINS: Short
 * val DOCUMENT_POSITION_CONTAINED_BY: Short
 * val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 * }
 * }

/** Exposes the JavaScript [HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin
 *
 * public external abstract class HTMLMetaElement : HTMLElement {
 * open var name: String
 * open var httpEquiv: String
 * open var content: String
 * open var scheme: String
 * companion object {
 * val ELEMENT_NODE: Short
 * val ATTRIBUTE_NODE: Short
 * val TEXT_NODE: Short
 * val CDATA_SECTION_NODE: Short
 * val ENTITY_REFERENCE_NODE: Short
 * val ENTITY_NODE: Short
 * val PROCESSING_INSTRUCTION_NODE: Short
 * val COMMENT_NODE: Short
 * val DOCUMENT_NODE: Short
 * val DOCUMENT_TYPE_NODE: Short
 * val DOCUMENT_FRAGMENT_NODE: Short
 * val NOTATION_NODE: Short
 * val DOCUMENT_POSITION_DISCONNECTED: Short
 * val DOCUMENT_POSITION_PRECEDING: Short
 * val DOCUMENT_POSITION_FOLLOWING: Short
 * val DOCUMENT_POSITION_CONTAINS: Short

```

```

 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin\n\npublic
external abstract class HTMLStyleElement : HTMLElement, LinkStyle {\n open var media: String\n open var
nonce: String\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement)
to Kotlin\n\npublic external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {\n
 open var text: String\n open var link: String\n open var vLink: String\n open var aLink: String\n
 open var
bgColor: String\n open var background: String\n\n companion object {\n val ELEMENT_NODE: Short\n
 val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
 val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLHeadingElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadingElement) to
Kotlin\n\npublic external abstract class HTMLHeadingElement : HTMLElement {\n open var align: String\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLParagraphElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParagraphElement) to Kotlin\n\n
public external abstract class HTMLParagraphElement : HTMLElement {\n open var align: String\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:

```

```

Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHRElement) to Kotlin\n *\npublic
external abstract class HTMLHRElement : HTMLElement {\n open var align: String\n open var color: String\n
open var noShade: Boolean\n open var size: String\n open var width: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin\n *\npublic
external abstract class HTMLPreElement : HTMLElement {\n open var width: Int\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin\n *\npublic
external abstract class HTMLQuoteElement : HTMLElement {\n open var cite: String\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLOListElement) to Kotlin\n *\npublic
external abstract class HTMLLOListElement : HTMLElement {\n open var reversed: Boolean\n open var start:
Int\n open var type: String\n open var compact: Boolean\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE:

```

```

Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n
val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement) to
Kotlin\n */\npublic external abstract class HTMLUListElement : HTMLInputElement {\n open var compact: Boolean\n
 open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLIElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLIElement) to Kotlin\n */\npublic
external abstract class HTMLLIElement : HTMLInputElement {\n open var value: Int\n open var type: String\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n
 val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val
NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLDListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDListElement) to
Kotlin\n */\npublic external abstract class HTMLDListElement : HTMLInputElement {\n open var compact:
Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE:
Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLDivElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDivElement) to Kotlin\n
*/\npublic external abstract class HTMLDivElement : HTMLInputElement {\n open var align: String\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE:
Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:

```



```

Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLAnchorElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAnchorElement) to
Kotlin\n */\npublic external abstract class HTMLAnchorElement : HTMLInputElement, HTMLHyperlinkElementUtils
{\n open var target: String\n open var download: String\n open var ping: String\n open var rel: String\n
open val relList: DOMTokenList\n open var hreflang: String\n open var type: String\n open var text: String\n
open var referrerPolicy: String\n open var coords: String\n open var charset:
String\n open var name: String\n open var rev: String\n open var shape: String\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin\n
*/\npublic external abstract class HTMLDataElement : HTMLInputElement {\n open var value: String\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin\n
*/\npublic external abstract class HTMLTimeElement : HTMLInputElement {\n open var dateTime: String\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement)
to Kotlin\n */\npublic external abstract class HTMLSpanElement : HTMLInputElement {\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

```





```

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement)
to Kotlin \n * \n public external abstract class HTMLObjectElement : HTMLElement { \n open var data: String \n
open var type: String \n open var typeMustMatch: Boolean \n open var name: String \n open var useMap:
String \n open val form: HTMLFormElement? \n open var width: String \n open var height: String \n open val
contentDocument: Document? \n open val contentWindow: Window? \n open val willValidate: Boolean \n open
val validity: ValidityState \n open val validationMessage: String \n open var align: String \n open var archive:
String \n open var code: String \n open var declare: Boolean \n open var hspace: Int \n open var standby:
String \n open var vspace: Int \n open var codeBase: String \n open var codeType: String \n open var border:
String \n fun getSVGDocument(): Document? \n fun checkValidity(): Boolean \n fun reportValidity():
Boolean \n fun setCustomValidity(error: String) \n \n
 companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n
 val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin \n
* \n public external abstract class HTMLParamElement : HTMLFormElement { \n open var name:
String \n open var value: String \n open var type: String \n open var valueType: String \n \n companion object
{ \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n
 val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement)
to Kotlin \n * \n public external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource,
TexImageSource { \n open var width: Int \n open var height: Int \n open val videoWidth: Int \n open val
videoHeight: Int \n open var poster: String \n open var playsInline: Boolean \n \n companion object { \n val
NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA:
Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val
HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n
 val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n
 val DOCUMENT_TYPE_NODE:

```

```

Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin\n
*/\npublic external abstract class HTMLAudioElement : HTMLMediaElement {\n companion object {\n val
NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA:
Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val
HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin\n
*/\npublic external abstract class HTMLTrackElement : HTMLMediaElement {\n open var kind: String\n open var src: String\n
open var srclang: String\n open var
label: String\n open var default: Boolean\n open val readyState: Short\n open val track: TextTrack\n\n
companion object {\n val NONE: Short\n val LOADING: Short\n val LOADED: Short\n val
ERROR: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin\n
*/\npublic external abstract class HTMLMediaElement : HTMLMediaElement {\n open val error: MediaError?\n open
var src: String\n open var srcObject: MediaProvider?\n open val currentSrc: String\n open var crossOrigin:
String?\n open val networkState: Short\n open var preload: String\n open val buffered: TimeRanges\n open
val readyState: Short\n open val seeking: Boolean\n open var currentTime: Double\n open val duration:
Double\n open val paused: Boolean\n open var defaultPlaybackRate: Double\n open var playbackRate:
Double\n open val played: TimeRanges\n open val seekable: TimeRanges\n open val ended: Boolean\n open
var autoplay: Boolean\n open var loop: Boolean\n open var controls: Boolean\n open var volume: Double\n
open var muted: Boolean\n open var defaultMuted: Boolean\n open val audioTracks:
AudioTrackList\n open val videoTracks: VideoTrackList\n open val textTracks: TextTrackList\n open val
mediaKeys: MediaKeys?\n open var onencrypted: ((Event) -> dynamic)?\n open var onwaitingforkey: ((Event) -
> dynamic)?\n fun load()\n fun canPlayType(type: String): CanPlayTypeResult\n fun fastSeek(time: Double)\n
fun getStartDate(): dynamic\n fun play(): Promise<Unit>\n fun pause()\n fun addTextTrack(kind:

```

```

TextTrackKind, label: String = definedExternally, language: String = definedExternally): TextTrack\n fun
setMediaKeys(mediaKeys: MediaKeys?): Promise<Unit>\n\n companion object {\n val
NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA:
Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val
HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n/**\n * Exposes the JavaScript
[MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin\n */\npublic external abstract
class MediaError {\n open val code: Short\n\n companion object {\n val MEDIA_ERR_ABORTED: Short\n
val MEDIA_ERR_NETWORK:
Short\n val MEDIA_ERR_DECODE: Short\n val MEDIA_ERR_SRC_NOT_SUPPORTED: Short\n
}\n}\n\n/**\n * Exposes the JavaScript
[AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin\n */\npublic external
abstract class AudioTrackList : EventTarget {\n open val length: Int\n open var onchange: ((Event) ->
dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) ->
dynamic)?\n fun getTrackById(id: String): AudioTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun AudioTrackList.get(index:
Int): AudioTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin\n */\npublic external abstract
class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n open val label: String\n
open val language: String\n open var enabled: Boolean\n open val sourceBuffer: SourceBuffer?\n}\n\n/**\n *
Exposes the JavaScript [VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to
Kotlin\n */\npublic external abstract class VideoTrackList : EventTarget {\n open val length: Int\n open val
selectedIndex: Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) ->
dynamic)?\n open var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
VideoTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun VideoTrackList.get(index:
Int): VideoTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin\n */\npublic external abstract
class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n
open val label: String\n open val language: String\n open var selected: Boolean\n open val sourceBuffer:
SourceBuffer?\n}\n\npublic external abstract class TextTrackList : EventTarget {\n open val length: Int\n open
var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var
onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
TextTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackList.get(index: Int):
TextTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin\n */\npublic external abstract

```

```

class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {
 open val kind: TextTrackKind
 open val label: String
 open val language: String
 open val id: String
 open val inBandMetadataTrackDispatchType: String
 open var mode: TextTrackMode
 open val cues: TextTrackCueList?
 open val activeCues: TextTrackCueList?
 open var oncuechange: ((Event) -> dynamic)?
 open val sourceBuffer: SourceBuffer?
 fun addCue(cue: TextTrackCue)
 fun removeCue(cue: TextTrackCue)
}

public external abstract class TextTrackCueList {
 open val length: Int
 fun getCueById(id: String): TextTrackCue?
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun TextTrackCueList.get(index: Int): TextTrackCue? = asDynamic()[index]

/** Exposes the JavaScript [TextTrackCue](https://developer.mozilla.org/en/docs/Web/API/TextTrackCue) to Kotlin */
public external abstract class TextTrackCue : EventTarget {
 open val track: TextTrack?
 open var id: String
 open var startTime: Double
 open var endTime: Double
 open var pauseOnExit: Boolean
 open var onenter: ((Event) -> dynamic)?
 open var onexit: ((Event) -> dynamic)?
}

/** Exposes the JavaScript [TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin */
public external abstract class TimeRanges {
 open val length: Int
 fun start(index: Int): Double
 fun end(index: Int): Double
}

/** Exposes the JavaScript [TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin */
public external open class TrackEvent(type: String, eventInitDict: TrackEventInit = definedExternally) : Event {
 open val track: UnionAudioTrackOrTextTrackOrVideoTrack?
}

companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
}

public external interface TrackEventInit : EventInit {
 var track: UnionAudioTrackOrTextTrackOrVideoTrack? /* = null */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun TrackEventInit(track: UnionAudioTrackOrTextTrackOrVideoTrack? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): TrackEventInit {
 val o = js("{}")
 o["track"] = track
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

/** Exposes the JavaScript [HTMLMapElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement) to Kotlin */
public external abstract class HTMLMapElement : HTMLElement {
 open val name: String
 open val areas: HTMLCollection
}

companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/** Exposes the JavaScript [HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement) to Kotlin */
public external abstract class HTMLAreaElement : HTMLElement, HTMLHyperlinkElementUtils {
 open val alt: String
 open val coords: String
 open val shape: String
 open val target: String
 open val download: String
 open val ping: String
 open val rel: String
 open val relList: DOMTokenList
 open var referrerPolicy: String
 open var noHref: Boolean
}

companion object {
 val ELEMENT_NODE:

```

```

Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin\n */\npublic
external abstract class HTMLTableElement : HTMLElement {\n open var caption:
HTMLTableCaptionElement?\n open var tHead:
HTMLTableSectionElement?\n open var tFoot: HTMLTableSectionElement?\n open val tBodies:
HTMLCollection\n open val rows: HTMLCollection\n open var align: String\n open var border: String\n
open var frame: String\n open var rules: String\n open var summary: String\n open var width: String\n
open
var bgColor: String\n open var cellPadding: String\n open var cellSpacing: String\n fun createCaption():
HTMLTableCaptionElement\n fun deleteCaption()\n fun createTHead(): HTMLTableSectionElement\n fun
deleteTHead()\n fun createTFoot(): HTMLTableSectionElement\n fun deleteTFoot()\n fun createTBody():
HTMLTableSectionElement\n fun insertRow(index: Int = definedExternally): HTMLTableRowElement\n fun
deleteRow(index: Int)\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n
 val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTableCaptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCaptionElement) to
Kotlin\n */\npublic external abstract class HTMLTableCaptionElement : HTMLElement {\n open var align:
String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTableColElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableColElement) to Kotlin\n */\npublic
external abstract class HTMLTableColElement : HTMLElement {\n open var span: Int\n open var
align: String\n open var ch: String\n open var chOff: String\n open var vAlign: String\n open var
width:
String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

```



```

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLTableSectionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableSectionElement) to
Kotlin \n * \n public external abstract class HTMLTableSectionElement : HTMLInputElement { \n open val rows:
HTMLCollection \n
 open var align: String \n open var ch: String \n open var chOff: String \n open var vAlign: String \n fun
insertRow(index: Int = definedExternally): HTMLInputElement \n fun deleteRow(index: Int) \n \n companion object
{ \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n
 val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val
ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE:
Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n
* Exposes the JavaScript
[HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin \n
* \n public external abstract class HTMLTableRowElement : HTMLInputElement { \n open val rowIndex: Int \n open
val sectionRowIndex: Int \n open val cells: HTMLCollection \n open var align: String \n open var ch: String \n
 open var chOff: String \n open var vAlign: String \n open var bgColor: String \n fun insertCell(index: Int =
definedExternally): HTMLInputElement \n fun deleteCell(index: Int) \n \n companion object { \n val
ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:
Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE:
Short \n val DOCUMENT_POSITION_DISCONNECTED: Short \n val
DOCUMENT_POSITION_PRECEDING: Short \n val DOCUMENT_POSITION_FOLLOWING: Short \n
 val DOCUMENT_POSITION_CONTAINS: Short \n val DOCUMENT_POSITION_CONTAINED_BY:
Short \n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short \n } \n} \n \n /** \n
* Exposes the JavaScript
[HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement)
to Kotlin \n * \n public external abstract class HTMLTableCellElement : HTMLInputElement { \n open var colSpan:
Int \n open var rowSpan: Int \n open var headers: String \n open val cellIndex: Int \n open var scope: String \n
 open var abbr: String \n open var align: String \n open var axis: String \n open var height: String \n open
var width: String \n open var ch: String \n open var chOff: String \n open var noWrap: Boolean \n open var vAlign:
String \n open var bgColor: String \n \n companion object { \n val ELEMENT_NODE:
Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val
CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE:
Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val
DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val
DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val
DOCUMENT_POSITION_DISCONNECTED: Short \n val DOCUMENT_POSITION_PRECEDING: Short \n
 val DOCUMENT_POSITION_FOLLOWING: Short \n val DOCUMENT_POSITION_CONTAINS: Short \n
 val DOCUMENT_POSITION_CONTAINED_BY: Short \n val

```



```

value: String\n open var valueAsDate: dynamic\n open var valueAsNumber: Double\n
open var width: Int\n open val willValidate: Boolean\n open val validity: ValidityState\n open val
validationMessage: String\n open val labels: NodeList\n open var selectionStart: Int?\n open var selectionEnd:
Int?\n open var selectionDirection: String?\n open var align: String\n open var useMap: String\n fun
stepUp(n: Int = definedExternally)\n fun stepDown(n: Int = definedExternally)\n fun checkValidity(): Boolean\n
fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n fun select()\n fun
setRangeText(replacement: String)\n fun setRangeText(replacement: String, start: Int, end: Int, selectionMode:
SelectionMode = definedExternally)\n fun setSelectionRange(start: Int, end: Int, direction: String =
definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin\n
*\npublic external abstract class HTMLButtonElement : HTMLElement {\n open var autofocus: Boolean\n
open var disabled: Boolean\n open val form: HTMLFormElement?\n open var formAction: String\n open var
formEnctype: String\n open var formMethod: String\n open
var formNoValidate: Boolean\n open var formTarget: String\n open var name: String\n open var type:
String\n open var value: String\n open var menu: HTMLMenuElement?\n open val willValidate: Boolean\n
open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n fun
checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin\n
*\npublic external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {\n open var
autocomplete: String\n open var autofocus: Boolean\n open var disabled: Boolean\n open val form:
HTMLFormElement?\n open var multiple: Boolean\n open var name: String\n open var required: Boolean\n
open var size: Int\n open val type: String\n open val options: HTMLOptionsCollection\n override var length:
Int\n open val selectedOptions: HTMLCollection\n open var selectedIndex: Int\n open var value: String\n
open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage:
String\n open val labels: NodeList\n fun namedItem(name: String): HTMLOptionElement?\n fun
add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n
fun remove(index: Int)\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun
setCustomValidity(error: String)\n override fun item(index: Int): Element?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:

```

```

Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n *
Exposes the JavaScript
[HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin\n
*/\npublic external abstract class HTMLDataListElement : HTMLElement {\n open val options:
HTMLCollection\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin\n
*/\npublic external abstract class HTMLOptGroupElement : HTMLElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n open var disabled: Boolean\n open var label:
String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin\n
*/\npublic external abstract class HTMLOptionElement : HTMLElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n open var disabled: Boolean\n open val form:
HTMLFormElement?\n
 open var label: String\n open var defaultSelected: Boolean\n open var selected: Boolean\n open var value:
String\n open var text: String\n open val index: Int\n\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
}
```

```

PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin\n
*\n\npublic external abstract class HTMLTextAreaElement : HTMLInputElement {\n open var autocomplete: String\n
open var autofocus: Boolean\n open var cols: Int\n open var dirName: String\n open var disabled: Boolean\n
open val form: HTMLFormElement?\n open var inputMode: String\n open var maxLength: Int\n open var
minLength: Int\n open var name: String\n open var placeholder: String\n open var readOnly: Boolean\n open
var required: Boolean\n open var rows: Int\n open var wrap: String\n open val type: String\n open var
defaultValue: String\n open var value: String\n open val textLength: Int\n open val willValidate: Boolean\n
open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n open var
selectionStart: Int?\n open var selectionEnd: Int?\n open var selectionDirection:
String?\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error:
String)\n fun select()\n fun setRangeText(replacement: String)\n fun setRangeText(replacement: String, start:
Int, end: Int, selectionMode: SelectionMode = definedExternally)\n fun setSelectionRange(start: Int, end: Int,
direction: String = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to
Kotlin\n
*\n\npublic external abstract class HTMLKeygenElement : HTMLInputElement {\n open var autofocus:
Boolean\n open var challenge: String\n open var disabled: Boolean\n open val form: HTMLFormElement?\n
open var keytype: String\n open var name: String\n open val type: String\n open val willValidate: Boolean\n
open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n fun
checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin\n
*\n\npublic external abstract class HTMLOutputElement : HTMLInputElement {\n open val htmlFor: DOMTokenList\n
open val form: HTMLFormElement?\n open var name: String\n open val type: String\n open var

```

```

defaultValue: String\n open
var value: String\n open val willValidate: Boolean\n open val validity: ValidityState\n open val
validationMessage: String\n open val labels: NodeList\n fun checkValidity(): Boolean\n fun reportValidity():
Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin\n
*\n\npublic external abstract class HTMLProgressElement : HTMLElement {\n open var value: Double\n open
var max: Double\n open val position: Double\n open val labels: NodeList\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to
Kotlin\n *\n\npublic external abstract class HTMLMeterElement : HTMLElement {\n open var value: Double\n
open var min: Double\n open var max: Double\n open var low: Double\n open var high: Double\n open var
optimum: Double\n open val labels: NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n
val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to
Kotlin\n *\n\npublic external abstract class HTMLFieldSetElement : HTMLElement {\n open var disabled:
Boolean\n open val form: HTMLFormElement?\n open var name: String\n open val type: String\n open val
elements: HTMLCollection\n open val willValidate: Boolean\n open val validity: ValidityState\n open val
validationMessage: String\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun
setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

```

```

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin \n
*/ \n public external abstract class HTMLLegendElement : HTMLElement { \n open val form:
HTMLFormElement? \n open var align: String \n \n companion
object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin \n
*/ \n public external
abstract class ValidityState { \n open val valueMissing: Boolean \n open val typeMismatch:
Boolean \n open val patternMismatch: Boolean \n open val tooLong: Boolean \n open val tooShort: Boolean \n
open val rangeUnderflow: Boolean \n open val rangeOverflow: Boolean \n open val stepMismatch: Boolean \n
open val badInput: Boolean \n open val customError: Boolean \n open val valid: Boolean \n} \n \n /** \n * Exposes
the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to
Kotlin \n
*/ \n public external abstract class HTMLDetailsElement : HTMLElement { \n open var open: Boolean \n
 companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n public external
abstract class HTMLMenuElement : HTMLElement { \n open var type: String \n open var label: String \n open
var compact: Boolean \n \n companion object { \n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n public external abstract class
HTMLMenuItemElement : HTMLElement { \n open var type: String \n open var label: String \n open var icon:
String \n open var disabled: Boolean \n open var checked: Boolean \n open var radiogroup: String \n open var
default: Boolean \n \n companion object { \n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:

```

```

Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external
open class RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {\n open val
relatedTarget: EventTarget?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
RelatedEventInit : EventInit {\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline fun RelatedEventInit(relatedTarget:
EventTarget?
= null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): RelatedEventInit {\n
val o = js(\{"{\}"})\n o["relatedTarget"] = relatedTarget\n o["bubbles"] = bubbles\n o["cancelable"] =
cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin\n
*/\n\npublic external abstract class HTMLDialogElement : HTMLInputElement {\n open var open: Boolean\n open var
returnValue: String\n fun show(anchor: UnionElementOrMouseEvent = definedExternally)\n fun
showModal(anchor: UnionElementOrMouseEvent = definedExternally)\n fun close(returnValue: String =
definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n
 val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin\n
*/\n\npublic external abstract class HTMLScriptElement : HTMLInputElement, HTMLOrSVGScriptElement {\n open var src:
String\n open var type: String\n open var charset: String\n open var async: Boolean\n open var defer:
Boolean\n open var crossOrigin: String?\n open var text: String\n open var nonce:
String\n open var event: String\n open var htmlFor: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin\n
*/

```



```

*\npublic external
abstract class HTMLTemplateElement : HTMLElement {\n open val content: DocumentFragment\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin\n */\n\n*\npublic external abstract class HTMLSlotElement : HTMLElement {\n open var name: String\n fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface AssignedNodesOptions {\n var flatten: Boolean? /* = false */\n fun get() = definedExternally\n fun set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {\n val o = js("{}")\n o["flatten"] = flatten\n return o\n}\n\n/**\n * Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin\n */\n\n*\npublic external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {\n open var width: Int\n open var height: Int\n fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?\n fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String\n fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface CanvasRenderingContext2DSettings {\n var alpha: Boolean? /* = true */\n fun get() = definedExternally\n fun set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {\n val o = js("{}")\n o["alpha"] = alpha\n return o\n}\n\n/**\n * Exposes the JavaScript

```

```
[CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to
Kotlin
*\npublic external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters,
CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion,
CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {\n
open val canvas: HTMLCanvasElement\n}\n\npublic external interface CanvasState {\n fun save()\n fun
restore()\n}\n\npublic external interface CanvasTransform {\n fun scale(x: Double, y: Double)\n fun
rotate(angle: Double)\n fun translate(x:
Double, y: Double)\n fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)\n fun
getTransform(): DOMMatrix\n fun setTransform(a: Double, b: Double, c: Double, d: Double, e: Double, f:
Double)\n fun setTransform(transform: dynamic = definedExternally)\n fun resetTransform()\n}\n\npublic
external interface CanvasCompositing {\n var globalAlpha: Double\n var globalCompositeOperation:
String\n}\n\npublic external interface CanvasImageSmoothing {\n var imageSmoothingEnabled: Boolean\n var
imageSmoothingQuality: ImageSmoothingQuality\n}\n\npublic external interface CanvasFillStrokeStyles {\n var
strokeStyle: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var fillStyle:
dynamic\n get() = definedExternally\n set(value) = definedExternally\n fun createLinearGradient(x0:
Double, y0: Double, x1: Double, y1: Double): CanvasGradient\n fun createRadialGradient(x0: Double, y0:
Double,
r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient\n fun createPattern(image:
CanvasImageSource, repetition: String): CanvasPattern?\n}\n\npublic external interface CanvasShadowStyles {\n
var shadowOffsetX: Double\n var shadowOffsetY: Double\n var shadowBlur: Double\n var shadowColor:
String\n}\n\npublic external interface CanvasFilters {\n var filter: String\n}\n\npublic external interface
CanvasRect {\n fun clearRect(x: Double, y: Double, w: Double, h: Double)\n fun fillRect(x: Double, y: Double,
w: Double, h: Double)\n fun strokeRect(x: Double, y: Double, w: Double, h: Double)\n}\n\npublic external
interface CanvasDrawPath {\n fun beginPath()\n fun fill(fillRule: CanvasFillRule = definedExternally)\n fun
fill(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n fun stroke()\n fun stroke(path: Path2D)\n
fun clip(fillRule: CanvasFillRule = definedExternally)\n fun clip(path: Path2D, fillRule: CanvasFillRule =
definedExternally)\n
fun resetClip()\n fun isPointInPath(x: Double, y: Double, fillRule: CanvasFillRule = definedExternally):
Boolean\n fun isPointInPath(path: Path2D, x: Double, y: Double, fillRule: CanvasFillRule = definedExternally):
Boolean\n fun isPointInStroke(x: Double, y: Double): Boolean\n fun isPointInStroke(path: Path2D, x: Double,
y: Double): Boolean\n}\n\npublic external interface CanvasUserInterface {\n fun drawFocusIfNeeded(element:
Element)\n fun drawFocusIfNeeded(path: Path2D, element: Element)\n fun scrollPathIntoView()\n fun
scrollPathIntoView(path: Path2D)\n}\n\npublic external interface CanvasText {\n fun fillText(text: String, x:
Double, y: Double, maxWidth: Double = definedExternally)\n fun strokeText(text: String, x: Double, y: Double,
maxWidth: Double = definedExternally)\n fun measureText(text: String): TextMetrics\n}\n\npublic external
interface CanvasDrawImage {\n fun drawImage(image: CanvasImageSource, dx: Double, dy: Double)\n
fun drawImage(image: CanvasImageSource, dx: Double, dy: Double, dw: Double, dh: Double)\n fun
drawImage(image: CanvasImageSource, sx: Double, sy: Double, sw: Double, sh: Double, dx: Double, dy: Double,
dw: Double, dh: Double)\n}\n\npublic external interface CanvasHitRegion {\n fun addHitRegion(options:
HitRegionOptions = definedExternally)\n fun removeHitRegion(id: String)\n fun clearHitRegions()\n}\n\npublic
external interface CanvasImageData {\n fun createImageData(sw: Double, sh: Double): ImageData\n fun
createImageData(imagedata: ImageData): ImageData\n fun getImageData(sx: Double, sy: Double, sw: Double, sh:
Double): ImageData\n fun putImageData(imagedata: ImageData, dx: Double, dy: Double)\n fun
putImageData(imagedata: ImageData, dx: Double, dy: Double, dirtyX: Double, dirtyY: Double, dirtyWidth: Double,
dirtyHeight: Double)\n}\n\npublic external interface CanvasPathDrawingStyles {\n var lineWidth: Double\n var
lineCap: CanvasLineCap\n
```

```

var lineJoin: CanvasLineJoin\n var miterLimit: Double\n var lineDashOffset: Double\n fun
setLineDash(segments: Array<Double>)\n fun getLineDash(): Array<Double>\n}\n\npublic external interface
CanvasTextDrawingStyles {\n var font: String\n var textAlign: CanvasTextAlign\n var textBaseline:
CanvasTextBaseline\n var direction: CanvasDirection\n}\n\npublic external interface CanvasPath {\n fun
closePath()\n fun moveTo(x: Double, y: Double)\n fun lineTo(x: Double, y: Double)\n fun
quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n fun bezierCurveTo(cp1x: Double, cp1y:
Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2:
Double, radius: Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY:
Double, rotation: Double)\n fun rect(x: Double, y: Double, w: Double, h: Double)\n fun arc(x: Double, y:
Double, radius: Double, startAngle: Double,
endAngle: Double, anticlockwise: Boolean = definedExternally)\n fun ellipse(x: Double, y: Double, radiusX:
Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean =
definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin\n */\n\npublic external
abstract class CanvasGradient {\n fun addColorStop(offset: Double, color: String)\n}\n\n/**\n * Exposes the
JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin\n */\n\npublic
external abstract class CanvasPattern {\n fun setTransform(transform: dynamic = definedExternally)\n}\n\n/**\n *
Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin\n
*/\n\npublic external abstract class TextMetrics {\n open val width: Double\n open val actualBoundingBoxLeft:
Double\n open val actualBoundingBoxRight: Double\n
open val fontBoundingBoxAscent: Double\n open val fontBoundingBoxDescent: Double\n open val
actualBoundingBoxAscent: Double\n open val actualBoundingBoxDescent: Double\n open val
emHeightAscent: Double\n open val emHeightDescent: Double\n open val hangingBaseline: Double\n open
val alphabeticBaseline: Double\n open val ideographicBaseline: Double\n}\n\npublic external interface
HitRegionOptions {\n var path: Path2D? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n var fillRule: CanvasFillRule? /* = CanvasFillRule.NONZERO */\n get() =
definedExternally\n set(value) = definedExternally\n var id: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var parentId: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var cursor: String? /* = "inherit" */\n get() =
definedExternally\n set(value)
= definedExternally\n var control: Element? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n var label: String? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n var role: String? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("\nINVISIBLE_REFERENCE",
"\nINVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun HitRegionOptions(path: Path2D? =
null, fillRule: CanvasFillRule? = CanvasFillRule.NONZERO, id: String? = "", parentId: String? = null, cursor:
String? = "inherit", control: Element? = null, label: String? = null, role: String? = null): HitRegionOptions {\n
val o = js("{}")\n o["path"] = path\n o["fillRule"] = fillRule\n o["id"] = id\n o["parentId"] = parentId\n
o["cursor"] = cursor\n o["control"] = control\n o["label"] = label\n o["role"] = role\n return
o}\n}\n\n/**\n * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin\n
*/\n\npublic external open class ImageData : ImageBitmapSource, TexImageSource {\n constructor(sw: Int, sh:
Int)\n constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)\n open val width: Int\n open
val height: Int\n open val data: Uint8ClampedArray\n}\n\n/**\n * Exposes the JavaScript
[Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin\n */\n\npublic external open class
Path2D() : CanvasPath {\n constructor(path: Path2D)\n constructor(paths: Array<Path2D>, fillRule:
CanvasFillRule = definedExternally)\n constructor(d: String)\n fun addPath(path: Path2D, transform: dynamic =
definedExternally)\n override fun closePath()\n override fun moveTo(x: Double, y: Double)\n override fun

```

```

lineTo(x: Double, y: Double)\n override fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n
 override fun bezierCurveTo(cp1x:
 Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n override fun arcTo(x1: Double, y1:
 Double, x2: Double, y2: Double, radius: Double)\n override fun arcTo(x1: Double, y1: Double, x2: Double, y2:
 Double, radiusX: Double, radiusY: Double, rotation: Double)\n override fun rect(x: Double, y: Double, w: Double,
 h: Double)\n override fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double,
 anticlockwise: Boolean /* = definedExternally */) \n override fun ellipse(x: Double, y: Double, radiusX: Double,
 radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* =
 definedExternally */) \n} \n\n/** \n * Exposes the JavaScript
[ImageBitmapRenderingContext](https://developer.mozilla.org/en/docs/Web/API/ImageBitmapRenderingContext)
to Kotlin \n * \n\npublic external abstract class ImageBitmapRenderingContext { \n open val canvas:
HTMLCanvasElement \n fun transferFromImageBitmap(bitmap:
 ImageBitmap?) \n} \n\npublic external interface ImageBitmapRenderingContextSettings { \n var alpha: Boolean? /*
= true */ \n get() = definedExternally \n set(value) =
 definedExternally \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",
 \"INVISIBLE_MEMBER\") \n@kotlin.internal.InlineOnly \npublic inline fun
ImageBitmapRenderingContextSettings(alpha: Boolean? = true): ImageBitmapRenderingContextSettings { \n val o
= js(\"({})\") \n o[\"alpha\"] = alpha \n return o \n} \n\n/** \n * Exposes the JavaScript
[CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin \n
* \n\npublic external abstract class CustomElementRegistry { \n fun define(name: String, constructor: () -> dynamic,
 options: ElementDefinitionOptions = definedExternally) \n fun get(name: String): Any? \n fun
 whenDefined(name: String): Promise<Unit> \n} \n\npublic external interface ElementDefinitionOptions { \n var
 extends: String? \n get() = definedExternally \n
 set(value) = definedExternally \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",
 \"INVISIBLE_MEMBER\") \n@kotlin.internal.InlineOnly \npublic inline fun ElementDefinitionOptions(extends:
 String? = undefined): ElementDefinitionOptions { \n val o = js(\"({})\") \n o[\"extends\"] = extends \n return
 o \n} \n\npublic external interface ElementContentEditable { \n var contentEditable: String \n val
 isContentEditable: Boolean \n} \n\n/** \n * Exposes the JavaScript
[DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer) to Kotlin \n * \n\npublic external
 abstract class DataTransfer { \n open var dropEffect: String \n open var effectAllowed: String \n open val items:
 DataTransferItemList \n open val types: Array<out String> \n open val files: FileList \n fun
 setDragImage(image: Element, x: Int, y: Int) \n fun getData(format: String): String \n fun setData(format: String,
 data: String) \n fun clearData(format: String = definedExternally) \n} \n\n/** \n * Exposes the
 JavaScript [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to
 Kotlin \n * \n\npublic external abstract class DataTransferItemList { \n open val length: Int \n fun add(data: String,
 type: String): DataTransferItem? \n fun add(data: File): DataTransferItem? \n fun remove(index: Int) \n fun
 clear() \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",
 \"INVISIBLE_MEMBER\") \n@kotlin.internal.InlineOnly \npublic inline operator fun
 DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index] \n} \n\n/** \n * Exposes the JavaScript
 [DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin \n * \n\npublic
 external abstract class DataTransferItem { \n open val kind: String \n open val type: String \n fun
 getAsString(_callback: ((String) -> Unit)?) \n fun getAsFile(): File? \n} \n\n/** \n * Exposes the JavaScript
 [DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin \n * \n\npublic external
 open class DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent { \n open val
 dataTransfer: DataTransfer? \n\n companion object { \n val NONE: Short \n val CAPTURING_PHASE:
 Short \n val AT_TARGET: Short \n val BUBBLING_PHASE: Short \n } \n} \n\npublic external interface
 DragEventInit : MouseEventInit { \n var dataTransfer: DataTransfer? /* = null */ \n get() = definedExternally \n
 set(value) = definedExternally \n} \n\n@Suppress(\"INVISIBLE_REFERENCE\",

```

```

\ "INVISIBLE_MEMBER"\n@kotlin.internal.InlineOnly\npublic inline fun DragEventInit(dataTransfer:
DataTransfer? = null, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0,
buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean?
= false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false,
modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false,
modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): DragEventInit {\n val o =
js("{}")\n o["dataTransfer"] = dataTransfer\n o["screenX"] = screenX\n o["screenY"] = screenY\n
o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n o["buttons"] = buttons\n
o["relatedTarget"] = relatedTarget\n o["region"] = region\n o["ctrlKey"] = ctrlKey\n o["shiftKey"] =
shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n o["modifierAltGraph"] =
modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] = modifierHyper\n o["modifierNumLock"]
= modifierNumLock\n o["modifierScrollLock"] = modifierScrollLock\n o["modifierSuper"] =
modifierSuper\n o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] =
modifierSymbolLock\n o["view"] = view\n o["detail"] = detail\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[Window](https://developer.mozilla.org/en/docs/Web/API/Window) to Kotlin\n */\npublic external abstract class
Window : EventTarget, GlobalEventHandlers, WindowEventHandlers, WindowOrWorkerGlobalScope,
WindowSessionStorage, WindowLocalStorage, GlobalPerformance, UnionMessagePortOrWindowProxy {\n open
val window: Window\n open val self: Window\n open val document: Document\n open val name: String\n
open val location: Location\n open val history: History\n open
val customElements: CustomElementRegistry\n open val locationbar: BarProp\n open val menubar: BarProp\n
open val personalbar: BarProp\n open val scrollbars: BarProp\n open val statusbar: BarProp\n open val
toolbar: BarProp\n open val status: String\n open val closed: Boolean\n open val frames: Window\n open val
length: Int\n open val top: Window\n open val opener: Any?\n open val parent: Window\n open val
frameElement: Element?\n open val navigator: Navigator\n open val applicationCache: ApplicationCache\n
open val external: External\n open val screen: Screen\n open val innerWidth: Int\n open val innerHeight: Int\n
open val scrollX: Double\n open val pageXOffset: Double\n open val scrollY: Double\n open val
pageYOffset: Double\n open val screenX: Int\n open val screenY: Int\n open val outerWidth: Int\n open val
outerHeight: Int\n open val devicePixelRatio: Double\n fun close()\n fun stop()\n
fun focus()\n fun blur()\n fun open(url: String = definedExternally, target: String = definedExternally,
features: String = definedExternally): Window?\n fun alert()\n fun alert(message: String)\n fun
confirm(message: String = definedExternally): Boolean\n fun prompt(message: String = definedExternally,
default: String = definedExternally): String?\n fun print()\n fun requestAnimationFrame(callback: (Double) ->
Unit): Int\n fun cancelAnimationFrame(handle: Int)\n fun postMessage(message: Any?, targetOrigin: String,
transfer: Array<dynamic> = definedExternally)\n fun captureEvents()\n fun releaseEvents()\n fun
matchMedia(query: String): MediaQueryList\n fun moveTo(x: Int, y: Int)\n fun moveBy(x: Int, y: Int)\n fun
resizeTo(x: Int, y: Int)\n fun resizeBy(x: Int, y: Int)\n fun scroll(options: ScrollToOptions = definedExternally)\n
fun scroll(x: Double, y: Double)\n fun scrollTo(options: ScrollToOptions = definedExternally)\n
fun scrollTo(x: Double, y: Double)\n fun scrollBy(options: ScrollToOptions = definedExternally)\n fun
scrollBy(x: Double, y: Double)\n fun getComputedStyle(elt: Element, pseudoElt: String? = definedExternally):
CSSStyleDeclaration\n}\n\n@Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER"\n@kotlin.internal.InlineOnly\npublic inline operator fun Window.get(name: String):
dynamic = asDynamic()[name]\n\npublic external abstract class BarProp {\n open val visible: Boolean\n}\n\n/**\n
* Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin\n */\npublic

```

```

external abstract class History {
 open val length: Int
 open var scrollRestoration: ScrollRestoration
 open val state: Any?
 fun go(delta: Int = definedExternally)
 fun back()
 fun forward()
 fun pushState(data: Any?, title: String, url: String? = definedExternally)
 fun replaceState(data: Any?, title: String, url: String? = definedExternally)
}

* Exposes the JavaScript [Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin
*
public external abstract class Location {
 open var href: String
 open val origin: String
 open var protocol: String
 open var host: String
 open var hostname: String
 open var port: String
 open var pathname: String
 open var search: String
 open var hash: String
 open val ancestorOrigins: Array<out String>
 fun assign(url: String)
 fun replace(url: String)
 fun reload()
}

* Exposes the JavaScript [PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin
*
public external open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {
 open val state: Any?
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external interface PopStateEventInit : EventInit {
 var state: Any? /* = null */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun PopStateEventInit(state: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PopStateEventInit {
 val o = js("{}")
 o["state"] = state
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

* Exposes the JavaScript [HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin
*
public external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) : Event {
 open val oldURL: String
 open val newURL: String
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external interface HashChangeEventInit : EventInit {
 var oldURL: String? /* = "" */
 get() = definedExternally
 set(value) = definedExternally
 var newURL: String? /* = "" */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun HashChangeEventInit(oldURL: String? = "", newURL: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): HashChangeEventInit {
 val o = js("{}")
 o["oldURL"] = oldURL
 o["newURL"] = newURL
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

* Exposes the JavaScript [PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin
*
public external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {
 open val persisted: Boolean
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external interface PageTransitionEventInit : EventInit {
 var persisted: Boolean? /* = false */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun PageTransitionEventInit(persisted: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PageTransitionEventInit {
 val o = js("{}")
 o["persisted"] = persisted
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

* Exposes the JavaScript [BeforeUnloadEvent](https://developer.mozilla.org/en/docs/Web/API/BeforeUnloadEvent) to Kotlin
*
public external open class BeforeUnloadEvent : Event {
 var returnValue: String
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external abstract class ApplicationCache : EventTarget {
 open val status: Short
 open var onchecking: ((Event) -> dynamic)?
 open var onerror:

```

```

((Event) -> dynamic)?\n open var onnoupdate: ((Event) -> dynamic)?\n open var ondownloading: ((Event) ->
dynamic)?\n open var onprogress: ((ProgressEvent) -> dynamic)?\n open var onupdateready: ((Event) ->
dynamic)?\n open var oncached: ((Event) -> dynamic)?\n open var onobsolete: ((Event) -> dynamic)?\n fun
update()\n fun abort()\n fun swapCache()\n\n companion object {\n val UNCACHED: Short\n val
IDLE: Short\n val CHECKING: Short\n val DOWNLOADING: Short\n val UPDATEREADY:
Short\n val OBSOLETE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[NavigatorOnLine](https://developer.mozilla.org/en/docs/Web/API/NavigatorOnLine) to Kotlin\n */\npublic
external interface NavigatorOnLine {\n val onLine: Boolean\n}\n\n/**\n * Exposes the JavaScript
[ErrorEvent](https://developer.mozilla.org/en/docs/Web/API/ErrorEvent) to Kotlin\n */\npublic external open class
ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {\n open val message: String\n
 open val filename: String\n open val lineno: Int\n open val colno: Int\n open val error: Any?\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ErrorEventInit : EventInit {\n var message:
String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var filename: String? /*
= \"\"
\n get() = definedExternally\n set(value) = definedExternally\n var lineno: Int? / = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var colno: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var error: Any? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ErrorEventInit(message: String? = \"\",
filename: String? = \"\", lineno: Int? = 0, colno: Int? = 0, error: Any? = null, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): ErrorEventInit {\n val o = js(\"({})\")\n o[\"message\"] =
message\n o[\"filename\"] = filename\n o[\"lineno\"] = lineno\n o[\"colno\"] = colno\n o[\"error\"] = error\n
o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n
 return o\n}\n\n/**\n * Exposes the JavaScript
[PromiseRejectionEvent](https://developer.mozilla.org/en/docs/Web/API/PromiseRejectionEvent) to Kotlin\n
*/\npublic external open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) :
Event {\n open val promise: Promise<Any?>\n open val reason: Any?\n\n companion object {\n val
NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface PromiseRejectionEventInit : EventInit {\n var
promise: Promise<Any?>?\n var reason: Any?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PromiseRejectionEventInit(promise:
Promise<Any?>?, reason: Any? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): PromiseRejectionEventInit
{\n val o = js(\"({})\")\n o[\"promise\"] = promise\n o[\"reason\"] = reason\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin\n
*/\npublic external interface GlobalEventHandlers {\n var onabort: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onblur: ((FocusEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oncancel: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oncanplay: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oncanplaythrough: ((Event) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n var onchange:
((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onclick:
((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onclose:
((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncontextmenu:
((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var

```

```

oncuechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ondblclick: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var ondrag: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ondragend: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ondragenter: ((DragEvent) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n var ondragexit: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondragleave: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondragover: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondragstart: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondrop: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ondurationchange: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onemptied: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onended: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value)
= definedExternally\n var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onfocus: ((FocusEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oninput: ((InputEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oninvalid: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onkeydown: ((KeyboardEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onkeypress: ((KeyboardEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onkeyup: ((KeyboardEvent)
-> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onload: ((Event) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onloadeddata: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onloadedmetadata: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onloadend: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onloadstart: ((ProgressEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onmousedown: ((MouseEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onmouseenter: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onmouseleave: ((MouseEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmousemove:
((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onmouseout: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onmouseover: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onmouseup: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onwheel: ((WheelEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onpause: ((Event) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onplay: ((Event) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onplaying: ((Event) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onprogress: ((ProgressEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onratechange: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onreset: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onresize: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onscroll: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onseeked: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onseeking: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onselect: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onshow: ((Event) -> dynamic)?\n get() =

```



```

definedExternally\n set(value) = definedExternally\n var onstalled: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n
 var onsubmit: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onsuspend: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ontimeupdate: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ontoggle: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onvolumechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onwaiting: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onlostpointercapture: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onpointerdown:
((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onpointermove: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onpointerup: ((PointerEvent) -> dynamic)?\n get() = definedExternally\n
set(value) = definedExternally\n var onpointercancel: ((PointerEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onpointerover: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onpointerout: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onpointerenter: ((PointerEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onpointerleave:
((PointerEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n/**\n
* Exposes the JavaScript
[WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEventHandlers) to Kotlin\n
*\n\npublic external interface WindowEventHandlers {\n var onafterprint: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onbeforeprint: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onbeforeunload: ((BeforeUnloadEvent) ->
String)?\n get() = definedExternally\n set(value) = definedExternally\n var onhashchange:
((HashChangeEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onlanguagechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onmessage: ((MessageEvent) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onoffline: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var ononline: ((Event) -> dynamic)?\n
get() = definedExternally\n set(value) = definedExternally\n var onpagehide: ((PageTransitionEvent) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onpageshow:
((PageTransitionEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var
onpopstate: ((PopStateEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n
var onrejectionhandled: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n var onstorage: ((StorageEvent) -> dynamic)?\n get() = definedExternally\n set(value)
= definedExternally\n var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onunload: ((Event) ->
dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n}\n\npublic external interface
DocumentAndElementEventHandlers {\n var oncopy: ((ClipboardEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var oncut: ((ClipboardEvent) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n var onpaste: ((ClipboardEvent) -> dynamic)?\n get()
= definedExternally\n set(value) = definedExternally\n}\n\n/**\n
* Exposes the JavaScript
[WindowOrWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WindowOrWorkerGlobalScope)
to Kotlin\n
*\n\npublic external interface WindowOrWorkerGlobalScope {\n val origin: String\n val caches:
CacheStorage\n fun btoa(data: String): String\n fun atob(data: String): String\n fun setTimeout(handler:
dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearTimeout(handle: Int =

```

```

definedExternally)\n
 fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun
clearInterval(handle: Int = definedExternally)\n fun createImageBitmap(image: ImageBitmapSource, options:
ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun createImageBitmap(image:
ImageBitmapSource, sx: Int, sy: Int, sw: Int, sh: Int, options: ImageBitmapOptions = definedExternally):
Promise<ImageBitmap>\n fun fetch(input: dynamic, init: RequestInit = definedExternally):
Promise<Response>\n}\n\n/**\n * Exposes the JavaScript
[Navigator](https://developer.mozilla.org/en/docs/Web/API/Navigator) to Kotlin\n */\n\npublic external abstract class
Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies,
NavigatorPlugins, NavigatorConcurrentHardware {\n open val clipboard: Clipboard\n open val mediaDevices:
MediaDevices\n open val maxTouchPoints: Int\n open val serviceWorker: ServiceWorkerContainer\n
 fun requestMediaKeySystemAccess(keySystem: String, supportedConfigurations:
Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>\n fun getUserMedia(constraints:
MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)\n fun
vibrate(pattern: dynamic): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorID](https://developer.mozilla.org/en/docs/Web/API/NavigatorID) to Kotlin\n */\n\npublic external interface
NavigatorID {\n val appCodeName: String\n val appName: String\n val appVersion: String\n val platform:
String\n val product: String\n val productSub: String\n val userAgent: String\n val vendor: String\n val
vendorSub: String\n val oscpu: String\n fun taintEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorLanguage](https://developer.mozilla.org/en/docs/Web/API/NavigatorLanguage) to Kotlin\n */\n\npublic
external interface NavigatorLanguage {\n val language: String\n
 val languages: Array<out String>\n}\n\npublic external interface NavigatorContentUtils {\n fun
registerProtocolHandler(scheme: String, url: String, title: String)\n fun registerContentHandler(mimeType: String,
url: String, title: String)\n fun isProtocolHandlerRegistered(scheme: String, url: String): String\n fun
isContentHandlerRegistered(mimeType: String, url: String): String\n fun unregisterProtocolHandler(scheme:
String, url: String)\n fun unregisterContentHandler(mimeType: String, url: String)\n}\n\npublic external interface
NavigatorCookies {\n val cookieEnabled: Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorPlugins](https://developer.mozilla.org/en/docs/Web/API/NavigatorPlugins) to Kotlin\n */\n\npublic
external interface NavigatorPlugins {\n val plugins: PluginArray\n val mimeTypes: MimeTypeArray\n fun
javaEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[PluginArray](https://developer.mozilla.org/en/docs/Web/API/PluginArray) to
Kotlin\n */\n\npublic external abstract class PluginArray : ItemArrayLike<Plugin> {\n fun refresh(reload: Boolean
= definedExternally)\n override fun item(index: Int): Plugin?\n fun namedItem(name: String):
Plugin?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun PluginArray.get(index: Int):
Plugin? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun PluginArray.get(name:
String): Plugin? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MimeTypeArray](https://developer.mozilla.org/en/docs/Web/API/MimeTypeArray) to Kotlin\n */\n\npublic external
abstract class MimeTypeArray : ItemArrayLike<MimeType> {\n override fun item(index: Int): MimeType?\n fun
namedItem(name: String): MimeType?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun MimeTypeArray.get(index:
Int): MimeType? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun MimeTypeArray.get(name:
String): MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin\n */\n\npublic external abstract class Plugin
: ItemArrayLike<MimeType> {\n open val name: String\n open val description: String\n open val filename:
String\n override fun item(index: Int): MimeType?\n fun namedItem(name: String):

```



```

composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin\n *\npublic external open
class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {\n open
val url: String\n open val withCredentials: Boolean\n open val readyState: Short\n var onopen: ((Event) ->
dynamic)?\n var onmessage: ((MessageEvent) -> dynamic)?\n var onerror: ((Event) -> dynamic)?\n fun
close()\n\n companion object {\n val CONNECTING: Short\n val OPEN: Short\n val CLOSED:
Short\n }\n}\n\npublic external interface EventSourceInit {\n var withCredentials: Boolean? /* = false *\n get() = definedExternally\n
 set(value) = definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventSourceInit(withCredentials:
Boolean? = false): EventSourceInit {\n val o = js(\\"({})")\n o[\"withCredentials\"] = withCredentials\n return
o\n}\n\n/**\n * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to
Kotlin\n *\npublic external open class WebSocket(url: String, protocols: dynamic = definedExternally) :
EventTarget {\n open val url: String\n open val readyState: Short\n open val bufferedAmount: Number\n var
onopen: ((Event) -> dynamic)?\n var onerror: ((Event) -> dynamic)?\n var onclose: ((Event) -> dynamic)?\n
open val extensions: String\n open val protocol: String\n var onmessage: ((MessageEvent) -> dynamic)?\n var
binaryType: BinaryType\n fun close(code: Short = definedExternally, reason: String = definedExternally)\n fun
send(data: String)\n fun send(data: Blob)\n fun send(data: ArrayBuffer)\n fun send(data:
ArrayBufferView)\n\n companion object {\n val CONNECTING: Short\n val OPEN: Short\n val
CLOSING: Short\n val CLOSED: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin\n *\npublic external open class
CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : Event {\n open val wasClean:
Boolean\n open val code: Short\n open val reason: String\n\n companion object {\n val NONE: Short\n
 val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface CloseEventInit : EventInit {\n var wasClean: Boolean? /* = false *\n get() =
definedExternally\n set(value) = definedExternally\n var code: Short? /* = 0 *\n get() =
definedExternally\n set(value)
= definedExternally\n var reason: String? /* = \"\" *\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CloseEventInit(wasClean: Boolean? =
false, code: Short? = 0, reason: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CloseEventInit {\n val o = js(\\"({})")\n o[\"wasClean\"] = wasClean\n o[\"code\"] = code\n
o[\"reason\"] = reason\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] =
composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin\n *\npublic external
open class MessageChannel {\n open val port1: MessagePort\n open val port2: MessagePort\n}\n\n/**\n *
Exposes the JavaScript [MessagePort](https://developer.mozilla.org/en/docs/Web/API/MessagePort)
to Kotlin\n *\npublic external abstract class MessagePort : EventTarget, UnionMessagePortOrWindowProxy,
UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n open var onmessage:
((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?, transfer: Array<dynamic> =
definedExternally)\n fun start()\n fun close()\n}\n\n/**\n * Exposes the JavaScript
[BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin\n *\npublic
external open class BroadcastChannel(name: String) : EventTarget {\n open val name: String\n var onmessage:
((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?)\n fun close()\n}\n\n/**\n * Exposes the
JavaScript [WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin\n
*\npublic external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope,
GlobalPerformance {\n open val self: WorkerGlobalScope\n open

```

```

val location: WorkerLocation\n open val navigator: WorkerNavigator\n open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n open var onlanguagechange: ((Event) -> dynamic)?\n open var onoffline: ((Event) -> dynamic)?\n open var ononline: ((Event) -> dynamic)?\n open var onrejectionhandled: ((Event) -> dynamic)?\n open var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n fun importScripts(vararg urls: String)\n}\n\n/**\n * Exposes the JavaScript
[DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to
Kotlin\n */\npublic external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {\n open var
onmessage: ((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?, transfer: Array<dynamic> =
definedExternally)\n fun close()\n}\n\n/**\n * Exposes the JavaScript
[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to
Kotlin\n */\npublic
external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {\n open val name: String\n open val
applicationCache: ApplicationCache\n open var onconnect: ((Event) -> dynamic)?\n fun close()\n}\n\n/**\n *
Exposes the JavaScript [AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to
Kotlin\n */\npublic external interface AbstractWorker {\n var onerror: ((Event) -> dynamic)?\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin\n */\npublic external open class
Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n var
onmessage: ((MessageEvent) -> dynamic)?\n override var onerror: ((Event) -> dynamic)?\n fun terminate()\n
fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\npublic external interface
WorkerOptions
{\n var type: WorkerType? /* = WorkerType.CLASSIC */\n get() = definedExternally\n set(value) =
definedExternally\n var credentials: RequestCredentials? /* = RequestCredentials.OMIT */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun WorkerOptions(type: WorkerType? =
WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {\n val o
= js(\"({})\")\n o[\"type\"] = type\n o[\"credentials\"] = credentials\n return o\n}\n\n/**\n * Exposes the
JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin\n */\npublic
external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions =
definedExternally) : EventTarget, AbstractWorker {\n open val port: MessagePort\n override var onerror:
((Event) -> dynamic)?\n}\n\n/**\n * Exposes the JavaScript
[NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to
Kotlin\n */\npublic external interface NavigatorConcurrentHardware {\n val hardwareConcurrency:
Number\n}\n\n/**\n * Exposes the JavaScript
[WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin\n */\npublic
external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine,
NavigatorConcurrentHardware {\n open val serviceWorker: ServiceWorkerContainer\n}\n\n/**\n * Exposes the
JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin\n */\n
public external abstract class WorkerLocation {\n open val href: String\n open val origin: String\n open val
protocol: String\n open val host: String\n open val hostname: String\n open val port: String\n open val
pathname: String\n open val search: String\n open val hash:
String\n}\n\n/**\n * Exposes the JavaScript [Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to
Kotlin\n */\npublic external abstract class Storage {\n open val length: Int\n fun key(index: Int): String?\n
fun removeItem(key: String)\n fun clear()\n fun getItem(key: String): String?\n fun setItem(key: String, value:
String)\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.get(key: String):
String? = asDynamic()[key]\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```

```

\@kotlin.internal.InlineOnly\npublic inline operator fun Storage.set(key: String, value:
String) { asDynamic()[key] = value }\n\n/*\n * Exposes the JavaScript
[WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin\n
\n\npublic external interface WindowSessionStorage {\n val sessionStorage: Storage\n}\n\n/\n * Exposes the
JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage)
to Kotlin\n
\n\npublic external interface WindowLocalStorage {\n val localStorage: Storage\n}\n\n/\n *
Exposes the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin\n
*\n\npublic external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) :
Event {\n open val key: String?\n open val oldValue: String?\n open val newValue: String?\n open val url:
String\n open val storageArea: Storage?\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
 }\n\n\npublic external interface StorageEventInit : EventInit {\n var key: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var oldValue: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var newValue: String? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n var url: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var storageArea:
Storage? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n@kotlin.internal.InlineOnly\npublic inline fun StorageEventInit(key: String? = null,
oldValue: String? = null, newValue: String? = null, url: String? = \\", storageArea: Storage? = null, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): StorageEventInit {\n val o =
js(\\"({})\\")\n o[\"key\"] = key\n o[\"oldValue\"] = oldValue\n o[\"newValue\"] = newValue\n o[\"url\"] =
url\n o[\"storageArea\"] = storageArea\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n
o[\"composed\"] = composed\n return o\n}\n\npublic
external abstract class HTMLAppletElement : HTMLDivElement {\n open var align: String\n open var alt: String\n
open var archive: String\n open var code: String\n open var codeBase: String\n open var height: String\n
open var hspace: Int\n open var name: String\n open var _object: String\n open var vspace: Int\n open var
width: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n\n/*\n * Exposes the JavaScript
[HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin\n
*\n\npublic external abstract class HTMLMarqueeElement : HTMLDivElement {\n open var behavior: String\n open
var bgColor: String\n open var direction: String\n open var height: String\n open var hspace: Int\n open
var loop: Int\n open var scrollAmount: Int\n open var scrollDelay: Int\n open var trueSpeed: Boolean\n open
var vspace: Int\n open var width: String\n open var onbounce: ((Event) -> dynamic)?\n open var onfinish:
((Event) -> dynamic)?\n open var onstart: ((Event) -> dynamic)?\n fun start()\n fun stop()\n\n companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

```



```

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\npublic external interface External
{\n fun AddSearchProvider()\n fun IsSearchProviderInstalled()\n}\n\npublic external interface EventInit {\n
var bubbles: Boolean?
/* = false */\n get() = definedExternally\n set(value) = definedExternally\n var cancelable: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var composed: Boolean? /* = false
*/\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventInit(bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): EventInit {\n val o = js("{}")\n o["bubbles"] =
bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the
JavaScript [CustomEvent](https://developer.mozilla.org/en/docs/Web/API/CustomEvent) to Kotlin\n */\n\npublic
external open class CustomEvent(type: String, eventInitDict: CustomEventInit = definedExternally) : Event {\n
open val detail: Any?\n fun initCustomEvent(type: String, bubbles:
Boolean, cancelable: Boolean, detail: Any?)\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface CustomEventInit : EventInit {\n var detail: Any? /* = null */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CustomEventInit(detail: Any? = null,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): CustomEventInit {\n val o
= js("{}")\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n
o["composed"] = composed\n return o\n}\n\npublic external interface EventListenerOptions {\n var capture:
Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventListenerOptions(capture:
Boolean? = false): EventListenerOptions {\n val o = js("{}")\n o["capture"] = capture\n return
o\n}\n\npublic external interface AddEventListenerOptions : EventListenerOptions {\n var passive: Boolean? /* =
false */\n get() = definedExternally\n set(value) = definedExternally\n var once: Boolean? /* = false */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun AddEventListenerOptions(passive:
Boolean? = false, once: Boolean? = false, capture: Boolean? = false): AddEventListenerOptions {\n val o =
js("{}")\n o["passive"] = passive\n o["once"] = once\n o["capture"] = capture\n return o\n}\n\npublic
external interface NonElementParentNode {\n fun getElementById(elementId: String): Element?\n}\n\n/**\n * Exposes the JavaScript
[DocumentOrShadowRoot](https://developer.mozilla.org/en/docs/Web/API/DocumentOrShadowRoot) to Kotlin\n */\n\npublic
external interface DocumentOrShadowRoot {\n val fullscreenElement: Element?\n get() =
definedExternally\n}\n\n/**\n * Exposes the JavaScript
[ParentNode](https://developer.mozilla.org/en/docs/Web/API/ParentNode) to Kotlin\n */\n\npublic external interface
ParentNode {\n val children: HTMLCollection\n val firstElementChild: Element?\n get() =
definedExternally\n val lastElementChild: Element?\n get() = definedExternally\n val childElementCount:
Int\n fun prepend(vararg nodes: dynamic)\n fun append(vararg nodes: dynamic)\n fun querySelector(selectors:
String): Element?\n fun querySelectorAll(selectors: String): NodeList\n}\n\n/**\n * Exposes the JavaScript
[NonDocumentTypeChildNode](https://developer.mozilla.org/en/docs/Web/API/NonDocumentTypeChildNode) to
Kotlin\n */\n\npublic external interface

```



```

NonDocumentTypeChildNode {\n val previousElementSibling: Element?\n get() = definedExternally\n val
nextElementSibling: Element?\n get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[ChildNode](https://developer.mozilla.org/en/docs/Web/API/ChildNode) to Kotlin\n */\n\npublic external interface
ChildNode {\n fun before(vararg nodes: dynamic)\n fun after(vararg nodes: dynamic)\n fun
replaceWith(vararg nodes: dynamic)\n fun remove()\n}\n\n/**\n * Exposes the JavaScript
[Slotable](https://developer.mozilla.org/en/docs/Web/API/Slotable) to Kotlin\n */\n\npublic external interface Slotable
{\n val assignedSlot: HTMLSlotElement?\n get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[NodeList](https://developer.mozilla.org/en/docs/Web/API/NodeList) to Kotlin\n */\n\npublic external abstract class
NodeList : ItemArrayLike<Node> {\n override fun item(index: Int):
Node?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic
inline operator fun NodeList.get(index: Int): Node? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[HTMLCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLCollection) to Kotlin\n */\n\npublic
external abstract class HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {\n
override fun item(index: Int): Element?\n fun namedItem(name: String):
Element?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun HTMLCollection.get(index:
Int): Element? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun HTMLCollection.get(name:
String): Element? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MutationObserver](https://developer.mozilla.org/en/docs/Web/API/MutationObserver) to Kotlin\n */\n\npublic
external open class MutationObserver(callback: (Array<MutationRecord>,
MutationObserver) -> Unit) {\n fun observe(target: Node, options: MutationObserverInit = definedExternally)\n
fun disconnect()\n fun takeRecords(): Array<MutationRecord>\n}\n\n/**\n * Exposes the JavaScript
[MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin\n
/\n\npublic external interface MutationObserverInit {\n var childList: Boolean? / = false */\n get() =
definedExternally\n set(value) = definedExternally\n var attributes: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var characterData: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var subtree: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var attributeOldValue: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var characterDataOldValue: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var attributeFilter: Array<String>?\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline fun MutationObserverInit(childList:
Boolean? = false, attributes: Boolean? = undefined, characterData: Boolean? = undefined, subtree: Boolean? = false,
attributeOldValue: Boolean? = undefined, characterDataOldValue: Boolean? = undefined, attributeFilter:
Array<String>? = undefined): MutationObserverInit {\n val o = js("{}")\n o["childList"] = childList\n
o["attributes"] = attributes\n o["characterData"] = characterData\n o["subtree"] = subtree\n
o["attributeOldValue"] = attributeOldValue\n o["characterDataOldValue"] = characterDataOldValue\n
o["attributeFilter"] = attributeFilter\n return o\n}\n\n/**\n * Exposes the JavaScript
[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord)
to Kotlin\n */\n\npublic external abstract class MutationRecord {\n open val type: String\n open val target:
Node\n open val addedNodes: NodeList\n open val removedNodes: NodeList\n open val previousSibling:
Node?\n open val nextSibling: Node?\n open val attributeName: String?\n open val attributeNamespace:
String?\n open val oldValue: String?\n}\n\n/**\n * Exposes the JavaScript
[Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin\n */\n\npublic external abstract class Node :
EventTarget {\n open val nodeType: Short\n open val nodeName: String\n open val baseURI: String\n open

```

```

val isConnected: Boolean\n open val ownerDocument: Document?\n open val parentNode: Node?\n open val
parentElement: Element?\n open val childNodes: NodeList\n open val firstChild: Node?\n open val lastChild:
Node?\n open val previousSibling: Node?\n open val nextSibling: Node?\n open var nodeValue: String?\n
 open var textContent: String?\n fun getRootNode(options: GetRootNodeOptions = definedExternally): Node\n
fun hasChildNodes(): Boolean\n fun normalize()\n fun cloneNode(deep: Boolean = definedExternally): Node\n
fun isEqualNode(otherNode: Node?): Boolean\n fun isSameNode(otherNode: Node?): Boolean\n fun
compareDocumentPosition(other: Node): Short\n fun contains(other: Node?): Boolean\n fun
lookupPrefix(namespace: String?): String?\n fun lookupNamespaceURI(prefix: String?): String?\n fun
isDefaultNamespace(namespace: String?): Boolean\n fun insertBefore(node: Node, child: Node?): Node\n fun
appendChild(node: Node): Node\n fun replaceChild(node: Node, child: Node): Node\n fun removeChild(child:
Node): Node\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n
 val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n public external interface
GetRootNodeOptions {\n var composed: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n }\n\n @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n @kotlin.internal.InlineOnly\n public inline fun GetRootNodeOptions(composed:
Boolean? = false): GetRootNodeOptions {\n val o = js(\"({})\")\n o[\"composed\"] = composed\n return
o\n }\n\n /**\n * Exposes the JavaScript
[Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin\n */\n public external open class
Document : Node, GlobalEventHandlers, DocumentAndElementEventHandlers, NonElementParentNode,
DocumentOrShadowRoot, ParentNode, GeometryUtils {\n open val implementation: DOMImplementation\n
open val URL: String\n open val documentURI: String\n open val origin: String\n open val compatMode:
String\n open val characterSet: String\n open val charset: String\n open val inputEncoding: String\n open val
contentType: String\n open val doctype: DocumentType?\n open val documentElement: Element?\n open val
location: Location?\n var domain: String\n open val referrer: String\n var cookie: String\n open val
lastModified: String\n open val readyState: DocumentReadyState\n var title: String\n var dir: String\n var
body: HTMLElement?\n open val head: HTMLHeadElement?\n open val images: HTMLCollection\n open val
embeds:
HTMLCollection\n open val plugins: HTMLCollection\n open val links: HTMLCollection\n open val forms:
HTMLCollection\n open val scripts: HTMLCollection\n open val currentScript: HTMLScriptElement?\n
 open val defaultView: Window?\n open val activeElement: Element?\n var designMode: String\n var
onreadystatechange: ((Event) -> dynamic)?\n var fgColor: String\n var linkColor: String\n var vlinkColor:
String\n var alinkColor: String\n var bgColor: String\n open val anchors: HTMLCollection\n open val
applets: HTMLCollection\n open val all: HTMLAllCollection\n open val scrollingElement: Element?\n open
val styleSheets: StyleSheetList\n open val rootElement: SVGSVGElement?\n open val fullscreenEnabled:
Boolean\n open val fullscreen: Boolean\n var onfullscreenchange: ((Event) -> dynamic)?\n var
onfullscreenerror: ((Event) -> dynamic)?\n override var onabort: ((Event) -> dynamic)?\n override var onblur:
((FocusEvent)
-> dynamic)?\n override var oncancel: ((Event) -> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n
 override var oncanplaythrough: ((Event) -> dynamic)?\n override var onchange: ((Event) -> dynamic)?\n
 override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose: ((Event) -> dynamic)?\n override var

```

```

oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var
ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var
ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var
ondragexit: ((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var
ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var
ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n
 override var onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var
onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n
 override var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var
onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n
 override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override
var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var
onloadend: ((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var
onmousedown: ((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n
 override var onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent)
-> dynamic)?\n override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover:
((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel:
((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) ->
dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) ->
dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n
 override var onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var
onseeked: ((Event) -> dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect:
((Event) -> dynamic)?\n override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) ->
dynamic)?\n override var onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n
 override var ontimeupdate: ((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override
var onvolumechange: ((Event) -> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n override var onpointerdown: ((PointerEvent) -> dynamic)?\n override var onpointermove:
((PointerEvent) -> dynamic)?\n override var onpointerup: ((PointerEvent) -> dynamic)?\n override var
onpointercancel: ((PointerEvent) -> dynamic)?\n override var onpointerover: ((PointerEvent) -> dynamic)?\n
 override var onpointerout: ((PointerEvent) -> dynamic)?\n override var onpointerenter: ((PointerEvent) ->
dynamic)?\n override var onpointerleave: ((PointerEvent) -> dynamic)?\n override var oncopy:
((ClipboardEvent) -> dynamic)?\n override var oncut: ((ClipboardEvent) -> dynamic)?\n override var onpaste:
((ClipboardEvent)
-> dynamic)?\n override val fullscreenElement: Element?\n override val children: HTMLCollection\n override
val firstElementChild: Element?\n override val lastElementChild: Element?\n override val childElementCount:
Int\n fun getElementsByTagName(qualifiedName: String): HTMLCollection\n fun
getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n fun
getElementsByTagName(className: String): HTMLCollection\n fun createElement(localName: String,
options: ElementCreationOptions = definedExternally): Element\n fun createElementNS(namespace: String?,
qualifiedName: String, options: ElementCreationOptions = definedExternally): Element\n fun
createDocumentFragment(): DocumentFragment\n fun createTextNode(data: String): Text\n fun
createCDATASection(data: String): CDATASection\n fun createComment(data: String): Comment\n fun
createProcessingInstruction(target: String, data: String): ProcessingInstruction\n fun importNode(node:
Node, deep: Boolean = definedExternally): Node\n fun adoptNode(node: Node): Node\n fun
createAttribute(localName: String): Attr\n fun createAttributeNS(namespace: String?, qualifiedName: String):
Attr\n fun createEvent(`interface`: String): Event\n fun createRange(): Range\n fun createNodeIterator(root:
Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): NodeIterator\n fun

```

```

createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)? =
definedExternally): NodeIterator\n fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter:
NodeFilter? = definedExternally): TreeWalker\n fun createTreeWalker(root: Node, whatToShow: Int =
definedExternally, filter: ((Node) -> Short)? = definedExternally): TreeWalker\n fun
getElementsByName(elementName: String): NodeList\n fun open(type: String = definedExternally, replace:
String = definedExternally): Document\n fun
open(url: String, name: String, features: String): Window\n fun close()\n fun write(vararg text: String)\n fun
writeln(vararg text: String)\n fun hasFocus(): Boolean\n fun execCommand(commandId: String, showUI:
Boolean = definedExternally, value: String = definedExternally): Boolean\n fun
queryCommandEnabled(commandId: String): Boolean\n fun queryCommandIndeterm(commandId: String):
Boolean\n fun queryCommandState(commandId: String): Boolean\n fun queryCommandSupported(commandId:
String): Boolean\n fun queryCommandValue(commandId: String): String\n fun clear()\n fun captureEvents()\n
fun releaseEvents()\n fun elementFromPoint(x: Double, y: Double): Element?\n fun elementsFromPoint(x:
Double, y: Double): Array<Element>\n fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?\n fun
createTouch(view: Window, target: EventTarget, identifier: Int, pageX: Int, pageY: Int, screenX: Int, screenY: Int):
Touch\n fun createTouchList(vararg
touches: Touch): TouchList\n fun exitFullscreen(): Promise<Unit>\n override fun getElementById(elementId:
String): Element?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes:
dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors:
String): NodeList\n override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */):
Array<DOMQuad>\n override fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Document.get(name: String):
dynamic = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin\n */\npublic external
open class XMLDocument
: Document {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

```

```

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\npublic external interface
ElementCreationOptions {\n var `is`: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\`INVISIBLE_REFERENCE\`,
\`INVISIBLE_MEMBER\`)\n@kotlin.internal.InlineOnly\npublic
inline fun ElementCreationOptions(`is`: String? = undefined): ElementCreationOptions {\n val o = js(\`({})\`)\n
o[\`is\`] = `is`\n return o\n}\n\n**\n * Exposes the JavaScript
[DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin\n
*\npublic external abstract class DOMImplementation {\n fun createDocumentType(qualifiedName: String,
publicId: String, systemId: String): DocumentType\n fun createDocument(namespace: String?, qualifiedName:
String, doctype: DocumentType? = definedExternally): XMLDocument\n fun createHTMLDocument(title: String
= definedExternally): Document\n fun hasFeature(): Boolean\n}\n\n**\n * Exposes the JavaScript
[DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin\n
*\npublic external
abstract class DocumentType : Node, ChildNode {\n open val name: String\n open val publicId: String\n open
val systemId: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n}\n\n**\n * Exposes the JavaScript
[DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin\n
*\npublic external open class DocumentFragment : Node, NonElementParentNode, ParentNode
{\n override val children: HTMLCollection\n override val firstElementChild: Element?\n override val
lastElementChild: Element?\n override val childElementCount: Int\n override fun getElementById(elementId:
String): Element?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes:
dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors:
String): NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n}\n\n**\n * Exposes the
JavaScript [ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin\n
*\npublic external open class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {\n open val mode:
ShadowRootMode\n open val host: Element\n override val fullscreenElement: Element?\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n

```

```

 val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**n * Exposes
the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin\n *
\npublic external
abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable,
GeometryUtils,
UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent,
UnionElementOrProcessingInstruction {n open val namespaceURI: String?\n open val prefix:
String?\n open
val localName: String\n open val tagName: String\n open var id: String\n open var className:
String\n open
val classList: DOMTokenList\n open var slot: String\n open val attributes: NamedNodeMap\n
open
val shadowRoot: ShadowRoot?\n open var scrollTop: Double\n open var scrollLeft: Double\n
open val
scrollWidth: Int\n open val scrollHeight: Int\n open val clientTop: Int\n open val clientLeft:
Int\n open val
clientWidth: Int\n open val clientHeight: Int\n open var innerHTML: String\n open var
outerHTML: String\n
fun hasAttributes(): Boolean\n fun getAttributeNames(): Array<String>\n fun
getAttribute(qualifiedName:
String): String?\n fun getAttributeNS(namespace: String?, localName: String): String?\n
fun
setAttribute(qualifiedName: String, value: String)\n fun setAttributeNS(namespace: String?,
qualifiedName:
String, value: String)\n fun removeAttribute(qualifiedName: String)\n fun
removeAttributeNS(namespace:
String?, localName: String)\n fun hasAttribute(qualifiedName: String): Boolean\n fun
hasAttributeNS(namespace: String?, localName: String): Boolean\n fun
getAttributeNode(qualifiedName: String):
Attr?\n fun
getAttributeNodeNS(namespace:
String?, localName: String): Attr?\n fun setAttributeNode(attr: Attr): Attr?\n fun
setAttributeNodeNS(attr: Attr):
Attr?\n fun removeAttributeNode(attr: Attr): Attr?\n fun attachShadow(init: ShadowRootInit):
ShadowRoot\n
fun closest(selectors: String): Element?\n fun matches(selectors: String): Boolean\n fun
webkitMatchesSelector(selectors: String): Boolean\n fun
getElementsByTagName(qualifiedName: String):
HTMLCollection\n fun
getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n fun
getElementsByClassName(classNames: String): HTMLCollection\n fun
insertAdjacentElement(when: String,
element: Element): Element?\n fun
insertAdjacentText(when: String, data: String)\n fun getClientRects():
Array<DOMRect>\n fun
getBoundingClientRect(): DOMRect\n fun scrollIntoView()\n fun
scrollIntoView(arg: dynamic)\n fun scroll(options: ScrollToOptions = definedExternally)\n
fun scroll(x: Double,
y:
Double)\n fun scrollTo(options: ScrollToOptions = definedExternally)\n fun
scrollTo(x: Double, y: Double)\n
fun scrollBy(options: ScrollToOptions = definedExternally)\n fun
scrollBy(x: Double, y: Double)\n fun
insertAdjacentHTML(position: String, text: String)\n fun
setPointerCapture(pointerId: Int)\n fun
releasePointerCapture(pointerId: Int)\n fun
hasPointerCapture(pointerId: Int): Boolean\n fun requestFullscreen():
Promise<Unit>\n\n companion object {n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\npublic external interface
ShadowRootInit {n var mode: ShadowRootMode?\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\",
\\"INVISIBLE_MEMBER\")n@kotlin.internal.InlineOnly\npublic inline fun ShadowRootInit(mode:
ShadowRootMode?): ShadowRootInit {n val o = js(\\"({})\")n o[\"mode\"] = mode\n return
o\n}\n\n/**n *
Exposes the JavaScript [NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to

```

```

Kotlin\n *\/\npublic external abstract class NamedNodeMap : ItemArrayLike<Attr> {\n fun
getNamedItemNS(namespace: String?, localName: String): Attr?\n fun setNamedItem(attr: Attr): Attr?\n fun
setNamedItemNS(attr: Attr): Attr?\n fun removeNamedItem(qualifiedName: String): Attr\n fun
removeNamedItemNS(namespace: String?,
localName: String): Attr\n override fun item(index: Int): Attr?\n fun getNamedItem(qualifiedName: String):
Attr?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun NamedNodeMap.get(index:
Int): Attr? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
NamedNodeMap.get(qualifiedName: String): Attr? = asDynamic()[qualifiedName]\n\n/**\n * Exposes the
JavaScript [Attr](https://developer.mozilla.org/en/docs/Web/API/Attr) to Kotlin\n *\/\npublic external abstract class
Attr : Node {\n open val namespaceURI: String?\n open val prefix: String?\n open val localName: String\n
open val name: String\n open var value: String\n open val ownerElement: Element?\n open val specified:
Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin\n *\/\npublic external
abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {\n open var data: String\n open
val length: Int\n fun substringData(offset: Int, count: Int): String\n fun appendData(data:
String)\n fun insertData(offset: Int, data: String)\n fun deleteData(offset: Int, count: Int)\n fun
replaceData(offset: Int, count: Int, data: String)\n\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[Text](https://developer.mozilla.org/en/docs/Web/API/Text)
to Kotlin\n *\/\npublic external open class Text(data: String = definedExternally) : CharacterData, Slotable,
GeometryUtils {\n open val wholeText: String\n override val assignedSlot: HTMLSlotElement?\n override val
previousElementSibling: Element?\n override val nextElementSibling: Element?\n fun splitText(offset: Int):
Text\n override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n
override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n override fun before(vararg nodes: dynamic)\n override fun after(vararg
nodes:

```

```

dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin\n
*/\npublic external open class CDATASection : Text {\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin\n
*/\npublic external abstract
class ProcessingInstruction : CharacterData, LinkStyle, UnionElementOrProcessingInstruction {\n open val
target: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[Comment](https://developer.mozilla.org/en/docs/Web/API/Comment)
to Kotlin\n */\npublic external open class Comment(data: String = definedExternally) : CharacterData {\n
 override val previousElementSibling: Element?\n override val nextElementSibling: Element?\n override fun
before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override fun replaceWith(vararg
nodes: dynamic)\n override fun remove()\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes

```



the JavaScript [\[Range\]](https://developer.mozilla.org/en/docs/Web/API/Range)(<https://developer.mozilla.org/en/docs/Web/API/Range>) to Kotlin\n \*\npublic external open class Range {\n open val startContainer: Node\n open val startOffset: Int\n open val endContainer: Node\n open val endOffset: Int\n open val collapsed: Boolean\n open val commonAncestorContainer: Node\n fun setStart(node: Node, offset: Int)\n fun setEnd(node: Node, offset: Int)\n fun setStartBefore(node: Node)\n fun setStartAfter(node: Node)\n fun setEndBefore(node: Node)\n fun setEndAfter(node: Node)\n fun collapse(toStart: Boolean = definedExternally)\n fun selectNode(node: Node)\n fun selectNodeContents(node: Node)\n fun compareBoundaryPoints(how: Short, sourceRange: Range): Short\n fun deleteContents()\n fun extractContents(): DocumentFragment\n fun cloneContents(): DocumentFragment\n fun insertNode(node: Node)\n fun surroundContents(newParent: Node)\n fun cloneRange(): Range\n fun detach()\n fun isPointInRange(node: Node, offset: Int): Boolean\n fun comparePoint(node: Node, offset: Int): Short\n fun intersectsNode(node: Node): Boolean\n fun getClientRects(): Array<DOMRect>\n fun getBoundingClientRect(): DOMRect\n fun createContextualFragment(fragment: String): DocumentFragment\n\n companion object {\n val START\_TO\_START: Short\n val START\_TO\_END: Short\n val END\_TO\_END: Short\n val END\_TO\_START: Short\n }\n}\n\n/\*\*\n \* Exposes the JavaScript [\[NodeIterator\]](https://developer.mozilla.org/en/docs/Web/API/NodeIterator)(<https://developer.mozilla.org/en/docs/Web/API/NodeIterator>) to Kotlin\n \*\npublic external abstract class NodeIterator {\n open val root: Node\n open val referenceNode: Node\n open val pointerBeforeReferenceNode: Boolean\n open val whatToShow: Int\n open val filter: NodeFilter?\n fun nextNode(): Node?\n fun previousNode(): Node?\n fun detach()\n}\n\n/\*\*\n \* Exposes the JavaScript [\[TreeWalker\]](https://developer.mozilla.org/en/docs/Web/API/TreeWalker)(<https://developer.mozilla.org/en/docs/Web/API/TreeWalker>) to Kotlin\n \*\npublic external abstract class TreeWalker {\n open val root: Node\n open val whatToShow: Int\n open val filter: NodeFilter?\n open var currentNode: Node\n fun parentNode(): Node?\n fun firstChild(): Node?\n fun lastChild(): Node?\n fun previousSibling(): Node?\n fun nextSibling(): Node?\n fun previousNode(): Node?\n fun nextNode(): Node?\n}\n\n/\*\*\n \* Exposes the JavaScript [\[NodeFilter\]](https://developer.mozilla.org/en/docs/Web/API/NodeFilter)(<https://developer.mozilla.org/en/docs/Web/API/NodeFilter>) to Kotlin\n \*\n@Suppress("NESTED\_CLASS\_IN\_EXTERNAL\_INTERFACE")\npublic external interface NodeFilter {\n fun acceptNode(node: Node): Short\n\n companion object {\n val FILTER\_ACCEPT: Short\n val FILTER\_REJECT: Short\n val FILTER\_SKIP: Short\n val SHOW\_ALL: Int\n val SHOW\_ELEMENT: Int\n val SHOW\_ATTRIBUTE: Int\n val SHOW\_TEXT: Int\n val SHOW\_CDATA\_SECTION: Int\n val SHOW\_ENTITY\_REFERENCE: Int\n val SHOW\_ENTITY: Int\n val SHOW\_PROCESSING\_INSTRUCTION: Int\n val SHOW\_COMMENT: Int\n val SHOW\_DOCUMENT: Int\n val SHOW\_DOCUMENT\_TYPE: Int\n val SHOW\_DOCUMENT\_FRAGMENT: Int\n val SHOW\_NOTATION: Int\n }\n}\n\n/\*\*\n \* Exposes the JavaScript [\[DOMTokenList\]](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList)(<https://developer.mozilla.org/en/docs/Web/API/DOMTokenList>) to Kotlin\n \*\npublic external abstract class DOMTokenList : ItemArrayLike<String> {\n open var value: String\n fun contains(token: String): Boolean\n fun add(vararg tokens: String)\n fun remove(vararg tokens: String)\n fun toggle(token: String, force: Boolean = definedExternally): Boolean\n fun replace(token: String, newToken: String)\n fun supports(token: String): Boolean\n override fun item(index: Int): String?\n}\n\n@Suppress("INVISIBLE\_REFERENCE", "INVISIBLE\_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMTokenList.get(index: Int): String? = asDynamic()[index]\n\n/\*\*\n \* Exposes the JavaScript [\[DOMPointReadOnly\]](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly)(<https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly>) to Kotlin\n \*\npublic external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {\n open val x: Double\n open val y: Double\n open val z: Double\n open val w: Double\n fun matrixTransform(matrix: DOMMatrixReadOnly): DOMPoint\n}\n\n/\*\*\n \* Exposes the JavaScript [\[DOMPoint\]](https://developer.mozilla.org/en/docs/Web/API/DOMPoint)(<https://developer.mozilla.org/en/docs/Web/API/DOMPoint>) to Kotlin\n \*\npublic external open class DOMPoint : DOMPointReadOnly {\n constructor(point: DOMPointInit)\n constructor(x: Double =

```

definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)\n
override var x: Double\n override var y: Double\n override var z: Double\n override var w:
Double\n}\n\n/**\n
 * Exposes the JavaScript [DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to
Kotlin\n */\npublic external interface DOMPointInit {\n var x: Double? /* = 0.0 */\n get() =
definedExternally\n set(value) = definedExternally\n var y: Double? /* = 0.0 */\n get() =
definedExternally\n set(value) = definedExternally\n var z: Double? /* = 0.0 */\n get() =
definedExternally\n set(value) = definedExternally\n var w: Double? /* = 1.0 */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMPointInit(x: Double? = 0.0, y:
Double? = 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {\n val o = js(\"({})\")\n o[\"x\"] = x\n
o[\"y\"] = y\n o[\"z\"] = z\n o[\"w\"] = w\n return o\n}\n\n/**\n * Exposes the JavaScript
[DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect)
to Kotlin\n */\npublic external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally,
width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {\n override var x:
Double\n override var y: Double\n override var width: Double\n override var height: Double\n}\n\n/**\n *
Exposes the JavaScript [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly)
to Kotlin\n */\npublic external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height:
Double) {\n open val x: Double\n open val y: Double\n open val width: Double\n open val height: Double\n
open val top: Double\n open val right: Double\n open val bottom: Double\n open val left: Double\n}\n\npublic
external interface DOMRectInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) =
definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n
set(value) = definedExternally\n var width: Double? /* = 0.0 */\n get() = definedExternally\n
set(value) = definedExternally\n var height: Double? /* = 0.0 */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMRectInit(x: Double? = 0.0, y:
Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {\n val o = js(\"({})\")\n o[\"x\"] =
x\n o[\"y\"] = y\n o[\"width\"] = width\n o[\"height\"] = height\n return o\n}\n\npublic external interface
DOMRectList : ItemArrayLike<DOMRect> {\n override fun item(index: Int):
DOMRect?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMRectList.get(index: Int):
DOMRect? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad)
to Kotlin\n */\npublic external open class DOMQuad {\n constructor(p1: DOMPointInit = definedExternally, p2:
DOMPointInit = definedExternally, p3: DOMPointInit = definedExternally, p4: DOMPointInit =
definedExternally)\n constructor(rect: DOMRectInit)\n open val p1: DOMPoint\n open val p2: DOMPoint\n
open val p3: DOMPoint\n open val p4: DOMPoint\n open val bounds: DOMRectReadOnly\n}\n\n/**\n *
Exposes the JavaScript
[DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin\n
*/\npublic external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {\n open val a: Double\n
open val b: Double\n open val c: Double\n open val d: Double\n open val e: Double\n open val f: Double\n
open val m11: Double\n open val m12: Double\n open val m13: Double\n open val m14: Double\n open val
m21: Double\n open val m22: Double\n open val m23: Double\n open val m24: Double\n open val m31:
Double\n open val m32: Double\n open val m33: Double\n open val m34: Double\n open val m41: Double\n
open val m42: Double\n open val m43: Double\n open val m44: Double\n open val is2D: Boolean\n open
val isIdentity: Boolean\n fun translate(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix\n
fun scale(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n
fun scale3d(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ:

```

```

Double = definedExternally): DOMMatrix\n fun scaleNonUniform(scaleX: Double, scaleY: Double =
definedExternally, scaleZ: Double = definedExternally, originX: Double = definedExternally, originY: Double =
definedExternally, originZ: Double = definedExternally): DOMMatrix\n fun rotate(angle: Double, originX:
Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n fun rotateFromVector(x:
Double, y: Double):
DOMMatrix\n fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix\n fun
skewX(sx: Double): DOMMatrix\n fun skewY(sy: Double): DOMMatrix\n fun multiply(other: DOMMatrix):
DOMMatrix\n fun flipX(): DOMMatrix\n fun flipY(): DOMMatrix\n fun inverse(): DOMMatrix\n fun
transformPoint(point: DOMPointInit = definedExternally): DOMPoint\n fun toFloat32Array(): Float32Array\n
fun toFloat64Array(): Float64Array\n}\n\n/**\n * Exposes the JavaScript
[DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin\n *\npublic external open
class DOMMatrix() : DOMMatrixReadOnly {\n constructor(transformList: String)\n constructor(other:
DOMMatrixReadOnly)\n constructor(array32: Float32Array)\n constructor(array64: Float64Array)\n
constructor(numberSequence: Array<Double>)\n override var a: Double\n override var b: Double\n override
var c: Double\n override var d: Double\n override var e: Double\n
override var f: Double\n override var m11: Double\n override var m12: Double\n override var m13:
Double\n override var m14: Double\n override var m21: Double\n override var m22: Double\n override var
m23: Double\n override var m24: Double\n override var m31: Double\n override var m32: Double\n override
var m33: Double\n override var m34: Double\n override var m41: Double\n override var m42: Double\n
override var m43: Double\n override var m44: Double\n fun multiplySelf(other: DOMMatrix): DOMMatrix\n
fun preMultiplySelf(other: DOMMatrix): DOMMatrix\n fun translateSelf(tx: Double, ty: Double, tz: Double =
definedExternally): DOMMatrix\n fun scaleSelf(scale: Double, originX: Double = definedExternally, originY:
Double = definedExternally): DOMMatrix\n fun scale3dSelf(scale: Double, originX: Double = definedExternally,
originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n fun
scaleNonUniformSelf(scaleX:
Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally, originX: Double =
definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n fun
rotateSelf(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally):
DOMMatrix\n fun rotateFromVectorSelf(x: Double, y: Double): DOMMatrix\n fun rotateAxisAngleSelf(x:
Double, y: Double, z: Double, angle: Double): DOMMatrix\n fun skewXSelf(sx: Double): DOMMatrix\n fun
skewYSelf(sy: Double): DOMMatrix\n fun invertSelf(): DOMMatrix\n fun setMatrixValue(transformList:
String): DOMMatrix\n}\n\npublic external interface ScrollOptions {\n var behavior: ScrollBehavior? /* =
ScrollBehavior.AUTO */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollOptions(behavior:
ScrollBehavior?
= ScrollBehavior.AUTO): ScrollOptions {\n val o = js(\"({})\")\n o[\"behavior\"] = behavior\n return
o\n}\n\n/**\n * Exposes the JavaScript
[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin\n *\npublic external
interface ScrollToOptions : ScrollOptions {\n var left: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var top: Double?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollToOptions(left: Double? =
undefined, top: Double? = undefined, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollToOptions {\n
val o = js(\"({})\")\n o[\"left\"] = left\n o[\"top\"] = top\n o[\"behavior\"] = behavior\n return o\n}\n\n/**\n *
Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to
Kotlin\n *\npublic

```

```

external abstract class MediaQueryList : EventTarget {
 open val media: String
 open val matches: Boolean
 open var onchange: ((Event) -> dynamic)?
 fun addListener(listener: EventListener?)
 fun addListener(listener: ((Event) -> Unit)?)
 fun removeListener(listener: EventListener?)
 fun removeListener(listener: ((Event) -> Unit)?)
}

/** Exposes the JavaScript [MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin
 *
 * public external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {
 open val media: String
 open val matches: Boolean
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
}

public external interface MediaQueryListEventInit : EventInit {
 var media: String? /* = "" */
 get() = definedExternally
 var matches: Boolean? /* = false */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun MediaQueryListEventInit(media: String? = "", matches: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MediaQueryListEventInit {
 val o = js("{}")
 o["media"] = media
 o["matches"] = matches
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
}

/** Exposes the JavaScript [Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin
 *
 * public external abstract class Screen {
 open val availWidth: Int
 open val availHeight: Int
 open val width: Int
 open val height: Int
 open val colorDepth: Int
 open val pixelDepth: Int
}

/** Exposes the JavaScript [CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin
 *
 * public external abstract class CaretPosition {
 open val offsetNode: Node
 open val offset: Int
 fun getClientRect(): DOMRect?
}

public external interface ScrollIntoViewOptions : ScrollOptions {
 var block: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */
 get() = definedExternally
 set(value) = definedExternally
 var inline: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ScrollIntoViewOptions(block: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, inline: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollIntoViewOptions {
 val o = js("{}")
 o["block"] = block
 o["inline"] = inline
 o["behavior"] = behavior
 return o
}

public external interface BoxQuadOptions {
 var box: CSSBoxType? /* = CSSBoxType.BORDER */
 get() = definedExternally
 set(value) = definedExternally
 var relativeTo: dynamic
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun BoxQuadOptions(box: CSSBoxType? = CSSBoxType.BORDER, relativeTo: dynamic = undefined): BoxQuadOptions {
 val o = js("{}")
 o["box"] = box
 o["relativeTo"] = relativeTo
 return o
}

public external interface ConvertCoordinateOptions {
 var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */
 get() = definedExternally
 set(value) = definedExternally
 var toBox: CSSBoxType? /* = CSSBoxType.BORDER */
 get() = definedExternally
 set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConvertCoordinateOptions(fromBox: CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER): ConvertCoordinateOptions {
 val o = js("{}")
 o["fromBox"] = fromBox
 o["toBox"] = toBox
 return o
}

/** Exposes the JavaScript [GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin
 *
 * public external interface GeometryUtils {
 fun getBoxQuads(options: BoxQuadOptions = definedExternally): Array<DOMQuad>
 fun convertQuadFromNode(quad: dynamic, from: dynamic, options:

```

```

ConvertCoordinateOptions = definedExternally): DOMQuad\n fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions =
definedExternally): DOMPoint\n}\n\n/**\n * Exposes the JavaScript
[Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin\n */\npublic external abstract class Touch
{\n open val identifier: Int\n open val target: EventTarget\n open val screenX: Int\n open val screenY: Int\n
open val clientX: Int\n open val clientY: Int\n open val pageX: Int\n open val pageY: Int\n open val region:
String?\n}\n\npublic external abstract class TouchList : ItemArrayLike<Touch> {\n override fun item(index: Int):
Touch?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TouchList.get(index: Int):
Touch? = asDynamic()[index]\n\npublic external open class TouchEvent : UIEvent {\n open val touches:
TouchList\n open val targetTouches: TouchList\n open val changedTouches: TouchList\n open val altKey:
Boolean\n open val metaKey: Boolean\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n\n
companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\n/**\n * Exposes the JavaScript [Image](https://developer.mozilla.org/en/docs/Web/API/Image) to
Kotlin\n */\npublic external open class Image(width: Int = definedExternally, height: Int = definedExternally) :
HTMLImageElement {\n override var onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) ->
dynamic)?\n override var onCancel: ((Event) -> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n
override var oncanplaythrough: ((Event) -> dynamic)?\n override var onChange: ((Event) -> dynamic)?\n
override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose: ((Event) -> dynamic)?\n override var
oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var
ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var
ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var
ondragexit: ((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var
ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var
ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var
onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror:
((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override
var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var
onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n
override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override
var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override
var onloadend: ((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var
onmousedown: ((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n
override var onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) ->
dynamic)?\n override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover:
((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel:
((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) ->
dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) ->
dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n
override var
onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked:
((Event) -> dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) ->
dynamic)?\n override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n
override var onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var
ontimeupdate: ((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var
onvolumechange: ((Event) -> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var

```

```

ongotpointercapture: ((PointerEvent) -> dynamic)?\n override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n override var onpointerdown: ((PointerEvent) -> dynamic)?\n override var onpointermove:
((PointerEvent) -> dynamic)?\n override var onpointerup: ((PointerEvent) -> dynamic)?\n override var
onpointercancel:
((PointerEvent) -> dynamic)?\n override var onpointerover: ((PointerEvent) -> dynamic)?\n override var
onpointerout: ((PointerEvent) -> dynamic)?\n override var onpointerenter: ((PointerEvent) -> dynamic)?\n
override var onpointerleave: ((PointerEvent) -> dynamic)?\n override var oncopy: ((ClipboardEvent) ->
dynamic)?\n override var oncut: ((ClipboardEvent) -> dynamic)?\n override var onpaste: ((ClipboardEvent) ->
dynamic)?\n override var contentEditable: String\n override val isContentEditable: Boolean\n override val
style: CSSStyleDeclaration\n override val children: HTMLCollection\n override val firstElementChild:
Element?\n override val lastElementChild: Element?\n override val childElementCount: Int\n override val
previousElementSibling: Element?\n override val nextElementSibling: Element?\n override val assignedSlot:
HTMLSlotElement?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes:
dynamic)\n
 override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String):
NodeList\n override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override
fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n override fun getBoxQuads(options:
BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n
 val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n public external open class
Audio(src: String = definedExternally) : HTMLAudioElement {\n override var onabort: ((Event) -> dynamic)?\n
override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel: ((Event) -> dynamic)?\n override var
oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough: ((Event) -> dynamic)?\n override var
onchange: ((Event)
-> dynamic)?\n override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose: ((Event) ->
dynamic)?\n override var oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange: ((Event) ->
dynamic)?\n override var ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag: ((DragEvent) ->
dynamic)?\n override var ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter: ((DragEvent) ->
dynamic)?\n override var ondragexit: ((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) ->
dynamic)?\n override var ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) ->
dynamic)?\n override var ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) ->
dynamic)?\n override var onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n
override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent)
-> dynamic)?\n override var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) ->
dynamic)?\n override var onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress:
((KeyboardEvent) -> dynamic)?\n override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload:

```

```

((Event) -> dynamic)?\n override var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata:
((Event) -> dynamic)?\n override var onloadend: ((Event) -> dynamic)?\n override var onloadstart:
((ProgressEvent) -> dynamic)?\n override var onmousedown: ((MouseEvent) -> dynamic)?\n override var
onmouseenter: ((MouseEvent) -> dynamic)?\n override var onmouseleave: ((MouseEvent) -> dynamic)?\n
override var onmousemove: ((MouseEvent) -> dynamic)?\n override var onmouseout: ((MouseEvent) ->
dynamic)?\n override var onmouseover: ((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent)
-> dynamic)?\n override var onwheel:
((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) ->
dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) ->
dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n
override var onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var
onseeked: ((Event) -> dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect:
((Event) -> dynamic)?\n override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) ->
dynamic)?\n override var onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n
override var ontimeupdate: ((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var
onvolumechange: ((Event) -> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n
 override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n override var onlostpointercapture:
((PointerEvent) -> dynamic)?\n override var onpointerdown: ((PointerEvent) -> dynamic)?\n override var
onpointermove: ((PointerEvent) -> dynamic)?\n override var onpointerup: ((PointerEvent) -> dynamic)?\n
override var onpointercancel: ((PointerEvent) -> dynamic)?\n override var onpointerover: ((PointerEvent) ->
dynamic)?\n override var onpointerout: ((PointerEvent) -> dynamic)?\n override var onpointerenter:
((PointerEvent) -> dynamic)?\n override var onpointerleave: ((PointerEvent) -> dynamic)?\n override var
oncopy: ((ClipboardEvent) -> dynamic)?\n override var oncut: ((ClipboardEvent) -> dynamic)?\n override var
onpaste: ((ClipboardEvent) -> dynamic)?\n override var contentEditable: String\n override val
isContentEditable: Boolean\n override val style: CSSStyleDeclaration\n override val children:
HTMLCollection\n override val
 firstElementChild: Element?\n override val lastElementChild: Element?\n override val childElementCount:
Int\n override val previousElementSibling: Element?\n override val nextElementSibling: Element?\n override
val assignedSlot: HTMLSlotElement?\n override fun prepend(vararg nodes: dynamic)\n override fun
append(vararg nodes: dynamic)\n override fun querySelector(selectors: String): Element?\n override fun
querySelectorAll(selectors: String): NodeList\n override fun before(vararg nodes: dynamic)\n override fun
after(vararg nodes: dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override
fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions
/* = definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object {\n
val NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA:
Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val
HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:

```

```

Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n } \n \n /** \n * Exposes the JavaScript
[Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin \n * \n public external open class
Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean =
definedExternally, selected: Boolean = definedExternally) : HTMLInputElement { \n override var onabort:
((Event) -> dynamic)? \n override var onblur: ((FocusEvent) -> dynamic)? \n override var oncancel: ((Event) ->
dynamic)? \n override var oncanplay: ((Event) -> dynamic)? \n override var oncanplaythrough: ((Event) ->
dynamic)? \n override var onchange: ((Event) -> dynamic)? \n override var onclick: ((MouseEvent) ->
dynamic)? \n override var onclose:
((Event) -> dynamic)? \n override var oncontextmenu: ((MouseEvent) -> dynamic)? \n override var
oncuechange: ((Event) -> dynamic)? \n override var ondblclick: ((MouseEvent) -> dynamic)? \n override var
ondrag: ((DragEvent) -> dynamic)? \n override var ondragend: ((DragEvent) -> dynamic)? \n override var
ondragenter: ((DragEvent) -> dynamic)? \n override var ondragexit: ((DragEvent) -> dynamic)? \n override var
ondragleave: ((DragEvent) -> dynamic)? \n override var ondragover: ((DragEvent) -> dynamic)? \n override var
ondragstart: ((DragEvent) -> dynamic)? \n override var ondrop: ((DragEvent) -> dynamic)? \n override var
ondurationchange: ((Event) -> dynamic)? \n override var onemptied: ((Event) -> dynamic)? \n override var
onended: ((Event) -> dynamic)? \n override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)? \n
override var onfocus: ((FocusEvent) -> dynamic)? \n override var oninput: ((InputEvent) -> dynamic)? \n override
var oninvalid:
((Event) -> dynamic)? \n override var onkeydown: ((KeyboardEvent) -> dynamic)? \n override var onkeypress:
((KeyboardEvent) -> dynamic)? \n override var onkeyup: ((KeyboardEvent) -> dynamic)? \n override var onload:
((Event) -> dynamic)? \n override var onloadeddata: ((Event) -> dynamic)? \n override var onloadedmetadata:
((Event) -> dynamic)? \n override var onloadend: ((Event) -> dynamic)? \n override var onloadstart:
((ProgressEvent) -> dynamic)? \n override var onmousedown: ((MouseEvent) -> dynamic)? \n override var
onmouseenter: ((MouseEvent) -> dynamic)? \n override var onmouseleave: ((MouseEvent) -> dynamic)? \n
override var onmousemove: ((MouseEvent) -> dynamic)? \n override var onmouseout: ((MouseEvent) ->
dynamic)? \n override var onmouseover: ((MouseEvent) -> dynamic)? \n override var onmouseup: ((MouseEvent)
-> dynamic)? \n override var onwheel: ((WheelEvent) -> dynamic)? \n override var onpause: ((Event) ->
dynamic)? \n override
var onplay: ((Event) -> dynamic)? \n override var onplaying: ((Event) -> dynamic)? \n override var onprogress:
((ProgressEvent) -> dynamic)? \n override var onratechange: ((Event) -> dynamic)? \n override var onreset:
((Event) -> dynamic)? \n override var onresize: ((Event) -> dynamic)? \n override var onscroll: ((Event) ->
dynamic)? \n override var onseeked: ((Event) -> dynamic)? \n override var onseeking: ((Event) -> dynamic)? \n
override var onselect: ((Event) -> dynamic)? \n override var onshow: ((Event) -> dynamic)? \n override var
onstalled: ((Event) -> dynamic)? \n override var onsubmit: ((Event) -> dynamic)? \n override var onsuspend:
((Event) -> dynamic)? \n override var ontimeupdate: ((Event) -> dynamic)? \n override var ontoggle: ((Event) ->
dynamic)? \n override var onvolumechange: ((Event) -> dynamic)? \n override var onwaiting: ((Event) ->
dynamic)? \n override var ongotpointercapture: ((PointerEvent) -> dynamic)? \n override var
onlostpointercapture:
((PointerEvent) -> dynamic)? \n override var onpointerdown: ((PointerEvent) -> dynamic)? \n override var
onpointermove: ((PointerEvent) -> dynamic)? \n override var onpointerup: ((PointerEvent) -> dynamic)? \n
override var onpointercancel: ((PointerEvent) -> dynamic)? \n override var onpointerover: ((PointerEvent) ->
dynamic)? \n override var onpointerout: ((PointerEvent) -> dynamic)? \n override var onpointerenter:
((PointerEvent) -> dynamic)? \n override var onpointerleave: ((PointerEvent) -> dynamic)? \n override var
oncopy: ((ClipboardEvent) -> dynamic)? \n override var oncut: ((ClipboardEvent) -> dynamic)? \n override var
onpaste: ((ClipboardEvent) -> dynamic)? \n override var contentEditable: String \n override val

```



```

isContentEditable: Boolean\n override val style: CSSStyleDeclaration\n override val children:
HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild: Element?\n
override val
 childElementCount: Int\n override val previousElementSibling: Element?\n override val nextElementSibling:
Element?\n override val assignedSlot: HTMLSlotElement?\n override fun prepend(vararg nodes: dynamic)\n
override fun append(vararg nodes: dynamic)\n override fun querySelector(selectors: String): Element?\n
override fun querySelectorAll(selectors: String): NodeList\n override fun before(vararg nodes: dynamic)\n
override fun after(vararg nodes: dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun
remove()\n override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n
 override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun
convertPointFromNode(point: DOMPointInit,
 from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
 ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
 Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n public external interface
UnionElementOrHTMLCollection\n\n public external interface UnionElementOrRadioNodeList\n\n public
external interface UnionHTMLOptGroupElementOrHTMLOptionElement\n\n public external interface
UnionAudioTrackOrTextTrackOrVideoTrack\n\n public external interface UnionElementOrMouseEvent\n\n public
external interface UnionMessagePortOrWindowProxy\n\n public external interface MediaPlayer\n\n public
external interface RenderingContext\n\n public external interface HTMLOrSVGImageElement :
CanvasImageSource\n\n public external interface CanvasImageSource : ImageBitmapSource\n\n public external
interface ImageBitmapSource\n\n public external interface HTMLOrSVGScriptElement\n\n/* please, don't
implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\n public external
interface DocumentReadyState {\n companion object\n }\n\n public inline val
DocumentReadyState.Companion.LOADING: DocumentReadyState get() =
"loading".asDynamic().unsafeCast<DocumentReadyState>()\n\n public inline val
DocumentReadyState.Companion.INTERACTIVE: DocumentReadyState get() =
"interactive".asDynamic().unsafeCast<DocumentReadyState>()\n\n public
inline val DocumentReadyState.Companion.COMPLETE: DocumentReadyState get() =
"complete".asDynamic().unsafeCast<DocumentReadyState>()\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\n public external
interface CanPlayTypeResult {\n companion object\n }\n\n public inline val
CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() =
"".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n public inline val CanPlayTypeResult.Companion.MAYBE:
CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n public inline val
CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() =
"probably".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\n public external
interface TextTrackMode {\n companion object\n }\n\n public

```

```

inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() =
 "disabled".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.HIDDEN: TextTrackMode get() =
 "hidden".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.SHOWING: TextTrackMode get() =
 "showing".asDynamic().unsafeCast<TextTrackMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface TextTrackKind {\n companion object\n}\n\npublic inline val TextTrackKind.Companion.SUBTITLES:
TextTrackKind get() = "subtitles".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
 "captions".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.DESCRPTIONS: TextTrackKind get() =
 "descriptions".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CHAPTERS:
 TextTrackKind get() = "chapters".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
 "metadata".asDynamic().unsafeCast<TextTrackKind>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface SelectionMode {\n companion object\n}\n\npublic inline val SelectionMode.Companion.SELECT:
SelectionMode get() = "select".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.START: SelectionMode get() =
 "start".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val SelectionMode.Companion.END:
SelectionMode get() = "end".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.PRESERVE: SelectionMode get() =
 "preserve".asDynamic().unsafeCast<SelectionMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic
external interface CanvasFillRule {\n companion object\n}\n\npublic inline val
CanvasFillRule.Companion.NONZERO: CanvasFillRule get() =
 "nonzero".asDynamic().unsafeCast<CanvasFillRule>()\n\npublic inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() =
 "evenodd".asDynamic().unsafeCast<CanvasFillRule>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface ImageSmoothingQuality {\n companion object\n}\n\npublic inline val
ImageSmoothingQuality.Companion.LOW: ImageSmoothingQuality get() =
 "low".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.MEDIUM: ImageSmoothingQuality get() =
 "medium".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() =
 "high".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\n/*
please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanvasLineCap {\n companion object\n}\n\npublic inline val CanvasLineCap.Companion.BUTT:
CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val
CanvasLineCap.Companion.ROUND: CanvasLineCap get() =
 "round".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val CanvasLineCap.Companion.SQUARE:
CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()\n\n/* please, don't implement this
interface! *\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic
external interface CanvasLineJoin {\n companion object\n}\n\npublic inline val

```

```

CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() =
`round`.asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val CanvasLineJoin.Companion.BEVEL:
CanvasLineJoin get() = `bevel`.asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic
inline val CanvasLineJoin.Companion.MITER: CanvasLineJoin get() =
`miter`.asDynamic().unsafeCast<CanvasLineJoin>()\n\n/* please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface CanvasTextAlign {\n companion object\n}\n\npublic inline val CanvasTextAlign.Companion.START:
CanvasTextAlign get() = `start`.asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.END: CanvasTextAlign get() =
`end`.asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val CanvasTextAlign.Companion.LEFT:
CanvasTextAlign get() = `left`.asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() =
`right`.asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() =
`center`.asDynamic().unsafeCast<CanvasTextAlign>()\n\n/* please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface CanvasTextBaseline {\n companion object\n}\n\npublic inline val CanvasTextBaseline.Companion.TOP:
CanvasTextBaseline get() = `top`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() =
`hanging`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() =
`middle`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() =
`alphabetic`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() =
`ideographic`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() =
`bottom`.asDynamic().unsafeCast<CanvasTextBaseline>()\n\n/*
please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface CanvasDirection {\n companion object\n}\n\npublic inline val CanvasDirection.Companion.LTR:
CanvasDirection get() = `ltr`.asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val
CanvasDirection.Companion.RTL: CanvasDirection get() =
`rtl`.asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val CanvasDirection.Companion.INHERIT:
CanvasDirection get() = `inherit`.asDynamic().unsafeCast<CanvasDirection>()\n\n/* please, don't implement this
interface! *\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic
external interface ScrollRestoration {\n companion object\n}\n\npublic inline val
ScrollRestoration.Companion.AUTO: ScrollRestoration get() =
`auto`.asDynamic().unsafeCast<ScrollRestoration>()\n\npublic inline val
ScrollRestoration.Companion.MANUAL: ScrollRestoration get() =
`manual`.asDynamic().unsafeCast<ScrollRestoration>()\n\n/*
please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface ImageOrientation {\n companion object\n}\n\npublic inline val ImageOrientation.Companion.NONE:
ImageOrientation get() = `none`.asDynamic().unsafeCast<ImageOrientation>()\n\npublic inline val
ImageOrientation.Companion.FLIPY: ImageOrientation get() =
`flipY`.asDynamic().unsafeCast<ImageOrientation>()\n\n/* please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external

```

```

interface PremultiplyAlpha {\n companion object\n}\n\npublic inline val PremultiplyAlpha.Companion.NONE:
PremultiplyAlpha get() = \"none\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
\"premultiply\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.DEFAULT:
PremultiplyAlpha get() = \"default\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\n/* please, don't implement
this interface! */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ColorSpaceConversion {\n companion object\n}\n\npublic inline val
ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() =
\"none\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\npublic inline val
ColorSpaceConversion.Companion.DEFAULT: ColorSpaceConversion get() =
\"default\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ResizeQuality {\n companion object\n}\n\npublic inline val ResizeQuality.Companion.PIXELATED:
ResizeQuality get() = \"pixelated\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.LOW: ResizeQuality get() =
\"low\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic
inline val ResizeQuality.Companion.MEDIUM: ResizeQuality get() =
\"medium\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.HIGH:
ResizeQuality get() = \"high\".asDynamic().unsafeCast<ResizeQuality>()\n\n/* please, don't implement this
interface! */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface BinaryType {\n companion object\n}\n\npublic inline val BinaryType.Companion.BLOB:
BinaryType get() = \"blob\".asDynamic().unsafeCast<BinaryType>()\n\npublic inline val
BinaryType.Companion.ARRAYBUFFER: BinaryType get() =
\"arraybuffer\".asDynamic().unsafeCast<BinaryType>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface WorkerType {\n companion object\n}\n\npublic inline val WorkerType.Companion.CLASSIC:
WorkerType get() = \"classic\".asDynamic().unsafeCast<WorkerType>()\n\npublic
inline val WorkerType.Companion.MODULE: WorkerType get() =
\"module\".asDynamic().unsafeCast<WorkerType>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ShadowRootMode {\n companion object\n}\n\npublic inline val ShadowRootMode.Companion.OPEN:
ShadowRootMode get() = \"open\".asDynamic().unsafeCast<ShadowRootMode>()\n\npublic inline val
ShadowRootMode.Companion.CLOSED: ShadowRootMode get() =
\"closed\".asDynamic().unsafeCast<ShadowRootMode>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ScrollBehavior {\n companion object\n}\n\npublic inline val ScrollBehavior.Companion.AUTO:
ScrollBehavior get() = \"auto\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val
ScrollBehavior.Companion.INSTANT: ScrollBehavior get() =
\"instant\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic
inline val ScrollBehavior.Companion.SMOOTH: ScrollBehavior get() =
\"smooth\".asDynamic().unsafeCast<ScrollBehavior>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ScrollLogicalPosition {\n companion object\n}\n\npublic inline val
ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() =
\"start\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() =
\"center\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val

```

```

ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() =
 \"end\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() =
 \"nearest\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface CSSBoxType {\n companion object\n}\n\npublic inline val
CSSBoxType.Companion.MARGIN: CSSBoxType get() =
 \"margin\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.BORDER:
CSSBoxType get() = \"border\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.PADDING: CSSBoxType get() =
 \"padding\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.CONTENT:
CSSBoxType get() = \"content\".asDynamic().unsafeCast<CSSBoxType>()\", \"/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED,
DO NOT EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.fetch\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.files.*\nimport
org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript
[Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n *\n\npublic external open class
Headers(init: dynamic = definedExternally) {\n fun append(name: String, value: String)\n fun delete(name:
String)\n fun get(name: String): String?\n fun has(name: String): Boolean\n fun set(name: String, value:
String)\n}\n\n/**\n * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to
Kotlin\n *\n\npublic external interface Body {\n val bodyUsed: Boolean\n fun arrayBuffer():
Promise<ArrayBuffer>\n fun blob(): Promise<Blob>\n fun formData(): Promise<FormData>\n fun json():
Promise<Any?>\n fun text(): Promise<String>\n}\n\n/**\n * Exposes the JavaScript
[Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n *\n\npublic external open class
Request(input: dynamic, init: RequestInit = definedExternally) : Body {\n open val method: String\n open val
url: String\n open val headers: Headers\n open val type: RequestType\n open val destination:
RequestDestination\n open val referrer: String\n open val referrerPolicy: dynamic\n open val mode:
RequestMode\n open val credentials: RequestCredentials\n open val cache: RequestCache\n open val redirect:
RequestRedirect\n open val integrity: String\n open val keepalive: Boolean\n override val bodyUsed:
Boolean\n fun clone(): Request\n override fun arrayBuffer(): Promise<ArrayBuffer>\n override fun blob():
Promise<Blob>\n override fun formData(): Promise<FormData>\n override fun json(): Promise<Any?>\n override
fun text(): Promise<String>\n}\n\npublic external interface RequestInit {\n var method: String?\n get() =
definedExternally\n set(value) = definedExternally\n var headers: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var body: dynamic\n get() = definedExternally\n set(
value) = definedExternally\n var referrer: String?\n get() = definedExternally\n set(value) =
definedExternally\n var referrerPolicy: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var mode: RequestMode?\n get() = definedExternally\n set(value) =
definedExternally\n var credentials: RequestCredentials?\n get() = definedExternally\n set(value) =
definedExternally\n var cache: RequestCache?\n get() = definedExternally\n set(value) =
definedExternally\n var redirect: RequestRedirect?\n get() = definedExternally\n set(value) =
definedExternally\n var integrity: String?\n get() = definedExternally\n set(value) = definedExternally\n var
keepalive: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var window:
Any?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun RequestInit(method: String? =
undefined, headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy:
dynamic = undefined, mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache:

```

```

RequestCache? = undefined, redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive:
Boolean? = undefined, window: Any? = undefined): RequestInit {\n val o = js("{}")\n o["method"] =
method\n o["headers"] = headers\n o["body"] = body\n o["referrer"] = referrer\n o["referrerPolicy"] =
referrerPolicy\n o["mode"] = mode\n o["credentials"] = credentials\n o["cache"] = cache\n o["redirect"]
= redirect\n o["integrity"] = integrity\n o["keepalive"] = keepalive\n o["window"] = window\n return
o}\n\n/**\n * Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response)
to Kotlin\n */\npublic external open class Response(body: dynamic = definedExternally, init: ResponseInit =
definedExternally) : Body {\n open val type: ResponseType\n open val url: String\n open val redirected:
Boolean\n open val status: Short\n open val ok: Boolean\n open val statusText: String\n open val headers:
Headers\n open val body: dynamic\n open val trailer: Promise<Headers>\n override val bodyUsed: Boolean\n
fun clone(): Response\n override fun arrayBuffer(): Promise<ArrayBuffer>\n override fun blob():
Promise<Blob>\n override fun formData(): Promise<FormData>\n override fun json(): Promise<Any?>\n
override fun text(): Promise<String>\n\n companion object {\n fun error(): Response\n fun redirect(url:
String, status: Short = definedExternally): Response\n }\n\n public external interface ResponseInit {\n var
status: Short? /* = 200 */\n get() = definedExternally\n set(value) = definedExternally\n var statusText: String? /* = "OK" */\n get() = definedExternally\n
set(value) = definedExternally\n var headers: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n }\n\n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n @kotlin.internal.InlineOnly\n public inline fun ResponseInit(status: Short? = 200,
statusText: String? = "OK", headers: dynamic = undefined): ResponseInit {\n val o = js("{}")\n o["status"]
= status\n o["statusText"] = statusText\n o["headers"] = headers\n return o}\n\n /** please, don't implement
this interface! */\n @JsName("null")\n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n public
external interface RequestType {\n companion object\n }\n\n public inline val RequestType.Companion.EMPTY:
RequestType get() = ""\n\n public inline val
RequestType.Companion.AUDIO: RequestType get() =
"audio"\n\n public inline val RequestType.Companion.FONT: RequestType get() =
"font"\n\n public inline val RequestType.Companion.IMAGE:
RequestType get() = "image"\n\n public inline val
RequestType.Companion.SCRIPT: RequestType get() =
"script"\n\n public inline val RequestType.Companion.STYLE:
RequestType get() = "style"\n\n public inline val
RequestType.Companion.TRACK: RequestType get() =
"track"\n\n public inline val RequestType.Companion.VIDEO:
RequestType get() = "video"\n\n /** please, don't implement this
interface! */\n @JsName("null")\n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n public
external interface RequestDestination {\n companion object\n }\n\n public inline val
RequestDestination.Companion.EMPTY: RequestDestination
get() = ""\n\n public inline val
RequestDestination.Companion.DOCUMENT: RequestDestination get() =
"document"\n\n public inline val
RequestDestination.Companion.EMBED: RequestDestination get() =
"embed"\n\n public inline val
RequestDestination.Companion.FONT: RequestDestination get() =
"font"\n\n public inline val
RequestDestination.Companion.IMAGE: RequestDestination get() =
"image"\n\n public inline val
RequestDestination.Companion.MANIFEST: RequestDestination get() =

```

```

\"manifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MEDIA: RequestDestination get() =
\"media\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.OBJECT: RequestDestination get()
= \"object\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.REPORT: RequestDestination get() =
\"report\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SCRIPT: RequestDestination get() =
\"script\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SERVICEWORKER: RequestDestination get() =
\"serviceworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
\"sharedworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.STYLE: RequestDestination get() =
\"style\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.WORKER: RequestDestination get() =
\"worker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.XSLT: RequestDestination
get() = \"xslt\".asDynamic().unsafeCast<RequestDestination>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestMode {\n companion object\n}\n\npublic inline val RequestMode.Companion.NAVIGATE:
RequestMode get() = \"navigate\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.SAME_ORIGIN: RequestMode get() = \"same-
origin\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val RequestMode.Companion.NO_CORS:
RequestMode get() = \"no-cors\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.CORS: RequestMode get() = \"cors\".asDynamic().unsafeCast<RequestMode>()\n\n/*
please, don't implement this interface!
*/\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCredentials {\n companion object\n}\n\npublic inline val RequestCredentials.Companion.OMIT:
RequestCredentials
get() = \"omit\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = \"same-
origin\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.INCLUDE: RequestCredentials get() =
\"include\".asDynamic().unsafeCast<RequestCredentials>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCache {\n companion object\n}\n\npublic inline val RequestCache.Companion.DEFAULT:
RequestCache get() = \"default\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_STORE: RequestCache get() = \"no-
store\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.RELOAD:
RequestCache get() = \"reload\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_CACHE: RequestCache get()
= \"no-cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.FORCE_CACHE: RequestCache get() = \"force-
cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = \"only-if-
cached\".asDynamic().unsafeCast<RequestCache>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external

```





```

Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var sampleSize:
Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var
echoCancellation: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n
var autoGainControl: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n
var noiseSuppression: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n
var latency: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var
channelCount: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var
deviceId: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var
groupId: Boolean? /* = true */\n get() = definedExternally\n
 set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
MediaTrackSupportedConstraints(width: Boolean? = true, height: Boolean? = true, aspectRatio: Boolean? = true,
frameRate: Boolean? = true, facingMode: Boolean? = true, resizeMode: Boolean? = true, volume: Boolean? = true,
sampleRate: Boolean? = true, sampleSize: Boolean? = true, echoCancellation: Boolean? = true, autoGainControl:
Boolean? = true, noiseSuppression: Boolean? = true, latency: Boolean? = true, channelCount: Boolean? = true,
deviceId: Boolean? = true, groupId: Boolean? = true): MediaTrackSupportedConstraints {\n val o = js(\"({})\")\n
o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] =
frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n
o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"]
= sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] = autoGainControl\n
o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n
o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return o\n}\n\npublic external interface
MediaTrackCapabilities {\n var width: ULongRange?\n get() = definedExternally\n set(value) =
definedExternally\n var height: ULongRange?\n get() = definedExternally\n set(value) =
definedExternally\n var aspectRatio: DoubleRange?\n get() = definedExternally\n set(value) =
definedExternally\n var frameRate: DoubleRange?\n get() = definedExternally\n set(value) =
definedExternally\n var facingMode: Array<String>?\n get() = definedExternally\n set(value) =
definedExternally\n var resizeMode: Array<String>?\n get() = definedExternally\n set(value) =
definedExternally\n
 var volume: DoubleRange?\n get() = definedExternally\n set(value) = definedExternally\n var
sampleRate: ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var
sampleSize: ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var
echoCancellation: Array<Boolean>?\n get() = definedExternally\n set(value) = definedExternally\n var
autoGainControl: Array<Boolean>?\n get() = definedExternally\n set(value) = definedExternally\n var
noiseSuppression: Array<Boolean>?\n get() = definedExternally\n set(value) = definedExternally\n var
latency: DoubleRange?\n get() = definedExternally\n set(value) = definedExternally\n var channelCount:
ULongRange?\n get() = definedExternally\n set(value) = definedExternally\n var deviceId: String?\n
get() = definedExternally\n set(value) = definedExternally\n
 var groupId: String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun MediaTrackCapabilities(width:
ULongRange? = undefined, height: ULongRange? = undefined, aspectRatio: DoubleRange? = undefined,
frameRate: DoubleRange? = undefined, facingMode: Array<String>? = undefined, resizeMode: Array<String>? =
undefined, volume: DoubleRange? = undefined, sampleRate: ULongRange? = undefined, sampleSize:
ULongRange? = undefined, echoCancellation: Array<Boolean>? = undefined, autoGainControl: Array<Boolean>?
= undefined, noiseSuppression: Array<Boolean>? = undefined, latency: DoubleRange? = undefined, channelCount:
ULongRange? = undefined, deviceId: String? = undefined, groupId: String? = undefined): MediaTrackCapabilities
{\n val o = js(\"({})\")\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n

```

```

o["frameRate"]
= frameRate\n o["facingMode"] = facingMode\n o["resizeMode"] = resizeMode\n o["volume"] =
volume\n o["sampleRate"] = sampleRate\n o["sampleSize"] = sampleSize\n o["echoCancellation"] =
echoCancellation\n o["autoGainControl"] = autoGainControl\n o["noiseSuppression"] = noiseSuppression\n
o["latency"] = latency\n o["channelCount"] = channelCount\n o["deviceId"] = deviceId\n o["groupId"] =
groupId\n return o\n}\n\n**\n * Exposes the JavaScript
[MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin\n
*/\npublic external interface MediaTrackConstraints : MediaTrackConstraintSet {\n var advanced:
Array<MediaTrackConstraintSet>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraints(advanced:
Array<MediaTrackConstraintSet>?
= undefined, width: dynamic = undefined, height: dynamic = undefined, aspectRatio: dynamic = undefined,
frameRate: dynamic = undefined, facingMode: dynamic = undefined, resizeMode: dynamic = undefined, volume:
dynamic = undefined, sampleRate: dynamic = undefined, sampleSize: dynamic = undefined, echoCancellation:
dynamic = undefined, autoGainControl: dynamic = undefined, noiseSuppression: dynamic = undefined, latency:
dynamic = undefined, channelCount: dynamic = undefined, deviceId: dynamic = undefined, groupId: dynamic =
undefined): MediaTrackConstraints {\n val o = js("{}")\n o["advanced"] = advanced\n o["width"] =
width\n o["height"] = height\n o["aspectRatio"] = aspectRatio\n o["frameRate"] = frameRate\n
o["facingMode"] = facingMode\n o["resizeMode"] = resizeMode\n o["volume"] = volume\n
o["sampleRate"] = sampleRate\n o["sampleSize"] = sampleSize\n o["echoCancellation"] =
echoCancellation\n o["autoGainControl"] =
autoGainControl\n o["noiseSuppression"] = noiseSuppression\n o["latency"] = latency\n
o["channelCount"] = channelCount\n o["deviceId"] = deviceId\n o["groupId"] = groupId\n return
o\n}\n\npublic external interface MediaTrackConstraintSet {\n var width: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var height: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n var aspectRatio: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var frameRate: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var facingMode: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var resizeMode: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var volume: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
var sampleRate:
dynamic\n get() = definedExternally\n set(value) = definedExternally\n var sampleSize: dynamic\n
get() = definedExternally\n set(value) = definedExternally\n var echoCancellation: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var autoGainControl: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var noiseSuppression: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var latency: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n var channelCount: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var deviceId: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
var groupId: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic
inline fun MediaTrackConstraintSet(width: dynamic = undefined, height: dynamic = undefined, aspectRatio:
dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined, resizeMode: dynamic =
undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize: dynamic = undefined,
echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined, noiseSuppression: dynamic =
undefined, latency: dynamic = undefined, channelCount: dynamic = undefined, deviceId: dynamic = undefined,
groupId: dynamic = undefined): MediaTrackConstraintSet {\n val o = js("{}")\n o["width"] = width\n

```

```

o["height"] = height\n o["aspectRatio"] = aspectRatio\n o["frameRate"] = frameRate\n o["facingMode"] =
facingMode\n o["resizeMode"] = resizeMode\n o["volume"] = volume\n o["sampleRate"] =
sampleRate\n o["sampleSize"] = sampleSize\n o["echoCancellation"] = echoCancellation\n
o["autoGainControl"]
= autoGainControl\n o["noiseSuppression"] = noiseSuppression\n o["latency"] = latency\n
o["channelCount"] = channelCount\n o["deviceId"] = deviceId\n o["groupId"] = groupId\n return
o\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin\n *^npublic
external interface MediaTrackSettings {\n var width: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var height: Int?\n get() = definedExternally\n set(value) = definedExternally\n var
aspectRatio: Double?\n get() = definedExternally\n set(value) = definedExternally\n var frameRate:
Double?\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: String?\n
get() = definedExternally\n set(value) = definedExternally\n var resizeMode: String?\n get() =
definedExternally\n set(value)
= definedExternally\n var volume: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var sampleRate: Int?\n get() = definedExternally\n set(value) = definedExternally\n
var sampleSize: Int?\n get() = definedExternally\n set(value) = definedExternally\n var echoCancellation:
Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var autoGainControl: Boolean?\n
get() = definedExternally\n set(value) = definedExternally\n var noiseSuppression: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var latency: Double?\n get() = definedExternally\n
set(value) = definedExternally\n var channelCount: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var deviceId: String?\n get() = definedExternally\n set(value) = definedExternally\n
var groupId: String?\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackSettings(width: Int? =
undefined, height: Int? = undefined, aspectRatio: Double? = undefined, frameRate: Double? = undefined,
facingMode: String? = undefined, resizeMode: String? = undefined, volume: Double? = undefined, sampleRate: Int?
= undefined, sampleSize: Int? = undefined, echoCancellation: Boolean? = undefined, autoGainControl: Boolean? =
undefined, noiseSuppression: Boolean? = undefined, latency: Double? = undefined, channelCount: Int? = undefined,
deviceId: String? = undefined, groupId: String? = undefined): MediaTrackSettings {\n val o = js("{}")\n
o["width"] = width\n o["height"] = height\n o["aspectRatio"] = aspectRatio\n o["frameRate"] =
frameRate\n o["facingMode"] = facingMode\n o["resizeMode"] = resizeMode\n o["volume"] = volume\n
o["sampleRate"]
= sampleRate\n o["sampleSize"] = sampleSize\n o["echoCancellation"] = echoCancellation\n
o["autoGainControl"] = autoGainControl\n o["noiseSuppression"] = noiseSuppression\n o["latency"] =
latency\n o["channelCount"] = channelCount\n o["deviceId"] = deviceId\n o["groupId"] = groupId\n
return o\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin\n *^npublic
external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) :
Event {\n open val track: MediaStreamTrack\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface MediaStreamTrackEventInit : EventInit {\n var track:
MediaStreamTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic
inline fun MediaStreamTrackEventInit(track: MediaStreamTrack?, bubbles: Boolean? = false, cancelable: Boolean?
= false, composed: Boolean? = false): MediaStreamTrackEventInit {\n val o = js("{}")\n o["track"] =
track\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\npublic external open class OverconstrainedErrorEvent(type: String, eventInitDict:

```

```

OverconstrainedErrorEventInit) : Event {\n open val error: dynamic\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface OverconstrainedErrorEventInit : EventInit {\n var error: dynamic /* =
null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun OverconstrainedErrorEventInit(error:
dynamic = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
OverconstrainedErrorEventInit {\n val o = js(\"({})\")\n o[\"error\"] = error\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin\n */\npublic external
abstract class MediaDevices : EventTarget {\n open var ondevicechange: ((Event) -> dynamic)?\n fun
enumerateDevices(): Promise<Array<MediaDeviceInfo>>\n fun getSupportedConstraints():
MediaTrackSupportedConstraints\n fun getUserMedia(constraints: MediaStreamConstraints = definedExternally):
Promise<MediaStream>\n}\n\n/**\n * Exposes the JavaScript
[MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to Kotlin\n */\npublic
external abstract class MediaDeviceInfo {\n open val deviceId: String\n open val kind:
MediaDeviceKind\n open val label: String\n open val groupId: String\n fun toJSON(): dynamic\n}\n\npublic
external abstract class InputDeviceInfo : MediaDeviceInfo {\n fun getCapabilities():
MediaTrackCapabilities\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to Kotlin\n
/\npublic external interface MediaStreamConstraints {\n var video: dynamic / = false */\n get() =
definedExternally\n set(value) = definedExternally\n var audio: dynamic /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamConstraints(video:
dynamic = false, audio: dynamic = false): MediaStreamConstraints {\n val o = js(\"({})\")\n o[\"video\"] =
video\n o[\"audio\"] = audio\n return o\n}\n\npublic external interface ConstrainingPattern
{\n var onoverconstrained: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) =
definedExternally\n fun getCapabilities(): Capabilities\n fun getConstraints(): Constraints\n fun getSettings():
Settings\n fun applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>\n}\n\n/**\n *
Exposes the JavaScript [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n
*/\npublic external interface DoubleRange {\n var max: Double?\n get() = definedExternally\n set(value)
= definedExternally\n var min: Double?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleRange(max: Double? =
undefined, min: Double? = undefined): DoubleRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] =
min\n return o\n}\n\npublic external interface
ConstrainDoubleRange : DoubleRange {\n var exact: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var ideal: Double?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainDoubleRange(exact: Double?
= undefined, ideal: Double? = undefined, max: Double? = undefined, min: Double? = undefined):
ConstrainDoubleRange {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n o[\"max\"] =
max\n o[\"min\"] = min\n return o\n}\n\npublic external interface ULongRange {\n var max: Int?\n get() =
definedExternally\n set(value) = definedExternally\n var min: Int?\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange(max: Int?
= undefined, min: Int? = undefined): ULongRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] =
min\n return o\n}\n\npublic external interface ConstrainULongRange : ULongRange {\n var exact: Int?

```

```

get() = definedExternally\n set(value) = definedExternally\n var ideal: Int?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainULongRange(exact: Int? =
undefined, ideal: Int? = undefined, max: Int? = undefined, min: Int? = undefined): ConstrainULongRange {\n val o
= js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n o[\"max\"] = max\n o[\"min\"] = min\n return
o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to
Kotlin\n\n * \npublic external interface ConstrainBooleanParameters {\n var exact:
Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var ideal: Boolean?\n get()
= definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainBooleanParameters(exact:
Boolean? = undefined, ideal: Boolean? = undefined): ConstrainBooleanParameters {\n val o = js(\"({})\")\n
o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n return o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters)
to Kotlin\n\n * \npublic external interface ConstrainDOMStringParameters {\n var exact: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var ideal: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline fun ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined):
ConstrainDOMStringParameters {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n return
o\n}\n\npublic external interface Capabilities\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Capabilities(): Capabilities {\n val o
= js(\"({})\")\n return o\n}\n\npublic external interface Settings\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Settings(): Settings {\n val o =
js(\"({})\")\n return o\n}\n\npublic external interface ConstraintSet\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstraintSet(): ConstraintSet {\n
val o = js(\"({})\")\n return o\n}\n\npublic external interface Constraints : ConstraintSet {\n var advanced:
Array<ConstraintSet>?\n
 get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Constraints(advanced:
Array<ConstraintSet>? = undefined): Constraints {\n val o = js(\"({})\")\n o[\"advanced\"] = advanced\n
return o\n}\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaStreamTrackState {\n companion object\n}\n\npublic inline val
MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() =
\"live\".asDynamic().unsafeCast<MediaStreamTrackState>()\n\npublic inline val
MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() =
\"ended\".asDynamic().unsafeCast<MediaStreamTrackState>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface VideoFacingModeEnum {\n
 companion object\n}\n\npublic inline val VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum
get() = \"user\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.ENVIRONMENT: VideoFacingModeEnum get() =
\"environment\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() =
\"left\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() =

```



```

ReadyState.Companion.ENDED: ReadyState get() = \"ended\".asDynamic().unsafeCast<ReadyState>()\n\n/*
please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface EndOfStreamError {\n companion object\n}\n\npublic inline val
EndOfStreamError.Companion.NETWORK: EndOfStreamError get() =
\"network\".asDynamic().unsafeCast<EndOfStreamError>()\n\npublic inline val
EndOfStreamError.Companion.DECODE: EndOfStreamError get() =
\"decode\".asDynamic().unsafeCast<EndOfStreamError>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface AppendMode {\n companion object\n}\n\npublic inline val AppendMode.Companion.SEGMENTS:
AppendMode get() = \"segments\".asDynamic().unsafeCast<AppendMode>()\n\npublic inline val
AppendMode.Companion.SEQUENCE: AppendMode get() =
\"sequence\".asDynamic().unsafeCast<AppendMode>()\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt
file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for
details\n\npackage org.w3c.dom.pointerevents\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface PointerEventInit : MouseEventInit {\n
var pointerId: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var width:
Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /*
= 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var pressure: Float? /* = 0f */\n
get() = definedExternally\n set(value) = definedExternally\n var tangentialPressure: Float? /* = 0f */\n
get() = definedExternally\n set(value) = definedExternally\n var tiltX: Int? /* = 0 */\n get() =
definedExternally\n set(value) = definedExternally\n var tiltY:
Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var twist: Int? /* = 0 */\n
get() = definedExternally\n set(value) = definedExternally\n var pointerType: String? /* = \"\" */\n get() =
definedExternally\n set(value) = definedExternally\n var isPrimary: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PointerEventInit(pointerId: Int? = 0,
width: Double? = 1.0, height: Double? = 1.0, pressure: Float? = 0f, tangentialPressure: Float? = 0f, tiltX: Int? = 0,
tiltY: Int? = 0, twist: Int? = 0, pointerType: String? = \"\", isPrimary: Boolean? = false, screenX: Int? = 0, screenY:
Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? =
null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false,
modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): PointerEventInit {\n val o = js(\"({})\")\n o[\"pointerId\"] = pointerId\n o[\"width\"] = width\n
o[\"height\"] = height\n o[\"pressure\"] = pressure\n o[\"tangentialPressure\"] = tangentialPressure\n
o[\"tiltX\"] = tiltX\n o[\"tiltY\"] = tiltY\n o[\"twist\"] = twist\n o[\"pointerType\"] = pointerType\n
o[\"isPrimary\"] = isPrimary\n o[\"screenX\"] = screenX\n o[\"screenY\"] = screenY\n o[\"clientX\"]
= clientX\n o[\"clientY\"] = clientY\n o[\"button\"] = button\n o[\"buttons\"] = buttons\n o[\"relatedTarget\"]
= relatedTarget\n o[\"region\"] = region\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n
o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"] = modifierAltGraph\n
o[\"modifierCapsLock\"] = modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] =
modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n
o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n

```

```

o["modifierSymbol"] = modifierSymbol\n o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] =
view\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] =
composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent)
to Kotlin\n */\n\npublic external open class PointerEvent(type: String, eventInitDict: PointerEventInit =
definedExternally) : MouseEvent {\n open val pointerId: Int\n open val width: Double\n open val height:
Double\n open val pressure: Float\n open val tangentialPressure: Float\n open val tiltX: Int\n open val tiltY:
Int\n open val twist: Int\n open val pointerType: String\n open val isPrimary: Boolean\n\n companion object
{\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n See
github.com/kotlin/dukat for details\n\npackage org.w3c.dom.svg\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.css.*\n\n/**\n * Exposes the JavaScript
[SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin\n */\n\npublic external
abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {\n
 open val dataset: DOMStringMap\n open val ownerSVGElement: SVGSVGElement?\n open val
viewportElement: SVGElement?\n open var tabIndex: Int\n fun focus()\n fun blur()\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface
SVGBoundingBoxOptions {\n var fill: Boolean? /* = true */\n get() = definedExternally\n set(value) =
definedExternally\n var stroke: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var markers: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var clipped: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun SVGBoundingBoxOptions(fill:
Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean?
= false): SVGBoundingBoxOptions {\n val o = js("{}")\n o["fill"] = fill\n o["stroke"] = stroke\n
o["markers"] = markers\n o["clipped"] = clipped\n return o\n}\n\n/**\n * Exposes the JavaScript
[SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin\n */\n\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n open val transform:
SVGAnimatedTransformList\n fun getBBox(options: SVGBoundingBoxOptions = definedExternally):
DOMRect\n fun getCTM(): DOMMatrix?\n fun getScreenCTM(): DOMMatrix?\n\n companion object {\n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n

```



```

 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin\n
*\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {\n open val pathLength:
SVGAnimatedNumber\n fun isPointInFill(point: DOMPoint): Boolean\n fun isPointInStroke(point: DOMPoint):
Boolean\n fun getTotalLength(): Float\n fun getPointAtLength(distance: Float): DOMPoint\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin\n
*\npublic external
abstract class SVGNumber {\n open var value: Float\n}\n\n/**\n * Exposes the JavaScript
[SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin\n
*\npublic external abstract
class SVGLength {\n open val unitType: Short\n open
var value: Float\n open var valueInSpecifiedUnits: Float\n open var valueAsString: String\n fun
newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)\n fun convertToSpecifiedUnits(unitType:
Short)\n\n companion object {\n val SVG_LENGTHTYPE_UNKNOWN: Short\n val
SVG_LENGTHTYPE_NUMBER: Short\n val SVG_LENGTHTYPE_PERCENTAGE: Short\n val
SVG_LENGTHTYPE_EMS: Short\n val SVG_LENGTHTYPE_EXS: Short\n val
SVG_LENGTHTYPE_PX: Short\n val SVG_LENGTHTYPE_CM: Short\n val
SVG_LENGTHTYPE_MM: Short\n val SVG_LENGTHTYPE_IN: Short\n val SVG_LENGTHTYPE_PT:
Short\n val SVG_LENGTHTYPE_PC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin\n
*\npublic external abstract
class SVGAngle {\n open val unitType: Short\n open var value: Float\n open var valueInSpecifiedUnits:
Float\n open var valueAsString: String\n fun
newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)\n fun
convertToSpecifiedUnits(unitType: Short)\n\n companion object {\n val SVG_ANGLETYPE_UNKNOWN:
Short\n val SVG_ANGLETYPE_UNSPECIFIED: Short\n val SVG_ANGLETYPE_DEG: Short\n val
SVG_ANGLETYPE_RAD: Short\n val SVG_ANGLETYPE_GRAD: Short\n }\n}\n\npublic external abstract
class SVGNameList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun
initialize(newItem: dynamic): dynamic\n fun insertItemBefore(newItem: dynamic, index: Int): dynamic\n fun
replaceItem(newItem: dynamic, index: Int): dynamic\n fun removeItem(index: Int): dynamic\n fun
appendItem(newItem: dynamic): dynamic\n fun getItem(index: Int):
dynamic\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.get(index: Int):
dynamic = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.set(index: Int,
newItem: dynamic) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin\n
*\npublic external
abstract class SVGNumberList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun
initialize(newItem: SVGNumber): SVGNumber\n fun insertItemBefore(newItem: SVGNumber, index: Int):
SVGNumber\n fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber\n fun removeItem(index: Int):
SVGNumber\n}

```

```

SVGNumber\n fun appendItem(newItem: SVGNumber): SVGNumber\n fun getItem(index: Int):
SVGNumber\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNumberList.get(index:
Int): SVGNumber? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem
}\n\n/**\n * Exposes the JavaScript
[SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin\n */\npublic external
abstract class SVGLengthList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun
initialize(newItem: SVGLength): SVGLength\n fun insertItemBefore(newItem: SVGLength, index: Int):
SVGLength\n fun replaceItem(newItem: SVGLength, index: Int): SVGLength\n fun removeItem(index: Int):
SVGLength\n fun appendItem(newItem: SVGLength): SVGLength\n fun getItem(index: Int):
SVGLength\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.get(index:
Int): SVGLength? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.set(index:
Int, newItem: SVGLength) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin\n */\npublic external abstract class SVGAnimatedBoolean {\n open var baseVal: Boolean\n open val animVal:
Boolean\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to
Kotlin\n */\npublic external abstract class SVGAnimatedEnumeration {\n open var baseVal: Short\n open val
animVal: Short\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin\n */\npublic external abstract class SVGAnimatedInteger {\n open var baseVal: Int\n open val animVal:
Int\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin\n */\npublic external abstract class SVGAnimatedNumber
{\n open var baseVal: Float\n open val animVal: Float\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin\n */\npublic external abstract class SVGAnimatedLength {\n open val baseVal: SVGLength\n open val animVal:
SVGLength\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin\n */\npublic
external abstract class SVGAnimatedAngle {\n open val baseVal: SVGAngle\n open val animVal:
SVGAngle\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin\n */\npublic
external abstract class SVGAnimatedString {\n open var baseVal: String\n open val animVal: String\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect)
to Kotlin\n */\npublic external abstract class SVGAnimatedRect
{\n open val baseVal: DOMRect\n open val animVal: DOMRectReadOnly\n}\n\n/**\n * Exposes the
JavaScript
[SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList)
to Kotlin\n */\npublic external abstract class SVGAnimatedNumberList {\n open val baseVal: SVGNumberList\n
open val animVal: SVGNumberList\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin\n */\npublic external abstract class SVGAnimatedLengthList {\n open val baseVal: SVGLengthList\n open val
animVal: SVGLengthList\n}\n\n/**\n * Exposes the JavaScript
[SVGStringList](https://developer.mozilla.org/en/docs/Web/API/SVGStringList) to Kotlin\n */\npublic external
abstract class SVGStringList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun

```

```

initialize(newItem: String): String\n fun insertItemBefore(newItem: String, index: Int): String\n fun
replaceItem(newItem:
String, index: Int): String\n fun removeItem(index: Int): String\n fun appendItem(newItem: String): String\n
fun getItem(index: Int): String\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.get(index:
Int): String? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.set(index: Int,
newItem: String) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGUnitTypes](https://developer.mozilla.org/en/docs/Web/API/SVGUnitTypes) to Kotlin\n
*\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface SVGUnitTypes
{\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val
SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTTests](https://developer.mozilla.org/en/docs/Web/API/SVGTTests)
to Kotlin\n *\npublic external interface SVGTTests {\n val requiredExtensions: SVGStringList\n val
systemLanguage: SVGStringList\n}\n\npublic external interface SVGFitToViewBox {\n val viewBox:
SVGAnimatedRect\n val preserveAspectRatio: SVGAnimatedPreserveAspectRatio\n}\n\n/**\n * Exposes the
JavaScript [SVGZoomAndPan](https://developer.mozilla.org/en/docs/Web/API/SVGZoomAndPan) to Kotlin\n
*\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface
SVGZoomAndPan {\n var zoomAndPan: Short\n\n companion object {\n val
SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val
SVG_ZOOMANDPAN_MAGNIFY: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGURIReference](https://developer.mozilla.org/en/docs/Web/API/SVGURIReference) to Kotlin\n *\npublic
external interface SVGURIReference {\n val href: SVGAnimatedString\n}\n\n/**\n * Exposes the JavaScript
[SVGSVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGSVGElement)
to Kotlin\n *\npublic external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox,
SVGZoomAndPan, WindowEventHandlers {\n open val x: SVGAnimatedLength\n open val y:
SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open
var currentScale: Float\n open val currentTranslate: DOMPointReadOnly\n fun getIntersectionList(rect:
DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun getEnclosureList(rect:
DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun checkIntersection(element: SVGElement,
rect: DOMRectReadOnly): Boolean\n fun checkEnclosure(element: SVGElement, rect: DOMRectReadOnly):
Boolean\n fun deselectAll()\n fun createSVGNumber(): SVGNumber\n fun createSVGLength(): SVGLength\n
fun createSVGAngle(): SVGAngle\n fun createSVGPoint(): DOMPoint\n fun createSVGMatrix():
DOMMatrix\n fun createSVGRect(): DOMRect\n
fun createSVGTransform(): SVGTransform\n fun createSVGTransformFromMatrix(matrix:
DOMMatrixReadOnly): SVGTransform\n fun getElementById(elementId: String): Element\n fun
suspendRedraw(maxWaitMilliseconds: Int): Int\n fun unsuspendRedraw(suspendHandleID: Int)\n fun
unsuspendRedrawAll()\n fun forceRedraw()\n\n companion object {\n val
SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val
SVG_ZOOMANDPAN_MAGNIFY: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:

```

```

Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin\n *\npublic
external abstract class SVGGElement : SVGGraphicsElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS:
Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGUnknownElement : SVGGraphicsElement {\n companion object {\n val ELEMENT_NODE: Short\n
 val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
 val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin\n
*\npublic external abstract class SVGDefsElement : SVGGraphicsElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGDescElement : SVGElement {\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin\n

```

```

*

public external abstract class SVGMetadataElement : SVGElement {

 companion object {

 val ELEMENT_NODE: Short

 val ATTRIBUTE_NODE: Short

 val TEXT_NODE: Short

 val CDATA_SECTION_NODE: Short

 val ENTITY_REFERENCE_NODE: Short

 val ENTITY_NODE: Short

 val PROCESSING_INSTRUCTION_NODE: Short

 val COMMENT_NODE: Short

 val DOCUMENT_NODE: Short

 val DOCUMENT_TYPE_NODE: Short

 val DOCUMENT_FRAGMENT_NODE: Short

 val NOTATION_NODE: Short

 val DOCUMENT_POSITION_DISCONNECTED: Short

 val DOCUMENT_POSITION_PRECEDING: Short

 val DOCUMENT_POSITION_FOLLOWING: Short

 val DOCUMENT_POSITION_CONTAINS: Short

 val DOCUMENT_POSITION_CONTAINED_BY: Short

 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

 }

}

* Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin

public external abstract class SVGTitleElement : SVGElement {

 companion object {

 val ELEMENT_NODE: Short

 val ATTRIBUTE_NODE: Short

 val TEXT_NODE: Short

 val CDATA_SECTION_NODE: Short

 val ENTITY_REFERENCE_NODE: Short

 val ENTITY_NODE: Short

 val PROCESSING_INSTRUCTION_NODE: Short

 val COMMENT_NODE: Short

 val DOCUMENT_NODE: Short

 val DOCUMENT_TYPE_NODE: Short

 val DOCUMENT_FRAGMENT_NODE: Short

 val NOTATION_NODE: Short

 val DOCUMENT_POSITION_DISCONNECTED: Short

 val DOCUMENT_POSITION_PRECEDING: Short

 val DOCUMENT_POSITION_FOLLOWING: Short

 val DOCUMENT_POSITION_CONTAINS: Short

 val DOCUMENT_POSITION_CONTAINED_BY: Short

 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

 }

}

* Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin

public external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {

 companion object {

 val ELEMENT_NODE: Short

 val ATTRIBUTE_NODE: Short

 val TEXT_NODE: Short

 val CDATA_SECTION_NODE: Short

 val ENTITY_REFERENCE_NODE: Short

 val ENTITY_NODE: Short

 val PROCESSING_INSTRUCTION_NODE: Short

 val COMMENT_NODE: Short

 val DOCUMENT_NODE: Short

 val DOCUMENT_TYPE_NODE: Short

 val DOCUMENT_FRAGMENT_NODE: Short

 val NOTATION_NODE: Short

 val DOCUMENT_POSITION_DISCONNECTED: Short

 val DOCUMENT_POSITION_PRECEDING: Short

 val DOCUMENT_POSITION_FOLLOWING: Short

 val DOCUMENT_POSITION_CONTAINS: Short

 val DOCUMENT_POSITION_CONTAINED_BY: Short

 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

 }

}

* Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin

public external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {

 open val x: SVGAnimatedLength

 open val y: SVGAnimatedLength

 open val width: SVGAnimatedLength

 open val height: SVGAnimatedLength

 open val instanceRoot: SVGElement?

 open val animatedInstanceRoot: SVGElement?

 companion object {

 val ELEMENT_NODE: Short

 val ATTRIBUTE_NODE: Short

 val TEXT_NODE: Short

 val CDATA_SECTION_NODE: Short

 val ENTITY_REFERENCE_NODE: Short

 val ENTITY_NODE: Short

 val PROCESSING_INSTRUCTION_NODE: Short

 val COMMENT_NODE: Short

 val DOCUMENT_NODE: Short

 val DOCUMENT_TYPE_NODE: Short

 val DOCUMENT_FRAGMENT_NODE: Short

 val NOTATION_NODE: Short

 val DOCUMENT_POSITION_DISCONNECTED: Short

 val DOCUMENT_POSITION_PRECEDING: Short

 val DOCUMENT_POSITION_FOLLOWING: Short

 val DOCUMENT_POSITION_CONTAINS: Short

 val DOCUMENT_POSITION_CONTAINED_BY: Short

 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

 }

}

public external open class

```

```

SVGUseElementShadowRoot : ShadowRoot {
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

public external interface SVGElementInstance {
 val correspondingElement: SVGElement?
 get() = definedExternally
 val correspondingUseElement: SVGUseElement?
 get() = definedExternally
}

public external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {
 open val sourceAnimation: dynamic
}

/**
 * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement) to Kotlin
 */
public external abstract class SVGSwitchElement : SVGGraphicsElement {
 companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 }
}

public external interface GetSVGDocument {
 fun getSVGDocument(): Document
}

/**
 * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin
 */
public external abstract class SVGStyleElement : SVGElement, LinkStyle {
 open var type: String
 open var media: String
 open var title: String
}

companion object {
 val ELEMENT_NODE: Short
 val ATTRIBUTE_NODE: Short
 val TEXT_NODE: Short
 val CDATA_SECTION_NODE: Short
 val ENTITY_REFERENCE_NODE: Short
 val ENTITY_NODE: Short
 val PROCESSING_INSTRUCTION_NODE: Short
 val COMMENT_NODE: Short
 val DOCUMENT_NODE: Short
 val DOCUMENT_TYPE_NODE: Short
 val DOCUMENT_FRAGMENT_NODE: Short
 val NOTATION_NODE: Short
 val DOCUMENT_POSITION_DISCONNECTED: Short
 val DOCUMENT_POSITION_PRECEDING: Short
 val DOCUMENT_POSITION_FOLLOWING: Short
 val DOCUMENT_POSITION_CONTAINS: Short
 val DOCUMENT_POSITION_CONTAINED_BY: Short
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin
 */
public external abstract class SVGTransform {
 open val type: Short
 open val matrix: DOMMatrix
 open val angle: Float
 fun setMatrix(matrix: DOMMatrixReadOnly)
 fun setTranslate(tx: Float, ty: Float)
 fun setScale(sx: Float, sy: Float)
 fun setRotate(angle: Float, cx: Float, cy: Float)
 fun setSkewX(angle: Float)
 fun setSkewY(angle: Float)
}

companion object {
 val SVG_TRANSFORM_UNKNOWN: Short
 val SVG_TRANSFORM_MATRIX: Short
 val SVG_TRANSFORM_TRANSLATE: Short
 val SVG_TRANSFORM_SCALE: Short
 val SVG_TRANSFORM_ROTATE: Short
 val SVG_TRANSFORM_SKEWX: Short
 val SVG_TRANSFORM_SKEWY: Short
}

/**
 * Exposes the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin
 */
public external abstract class SVGTransformList {
 open val length: Int
 open val numberOfItems: Int
 fun

```

```

clear()\n fun initialize(newItem: SVGTransform): SVGTransform\n fun insertItemBefore(newItem:
SVGTransform, index: Int): SVGTransform\n fun replaceItem(newItem: SVGTransform, index: Int):
SVGTransform\n fun removeItem(index: Int): SVGTransform\n fun appendItem(newItem: SVGTransform):
SVGTransform\n fun createSVGTransformFromMatrix(matrix:
DOMMatrixReadOnly): SVGTransform\n fun consolidate(): SVGTransform?\n fun getItem(index: Int):
SVGTransform\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.get(index:
Int): SVGTransform? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.set(index:
Int, newItem: SVGTransform) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to
Kotlin\n */\npublic external abstract class SVGAnimatedTransformList {\n open val baseVal:
SVGTransformList\n open val animVal: SVGTransformList\n}\n\n/**\n * Exposes the JavaScript
[SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin\n
*/\npublic external abstract class SVGPreserveAspectRatio
{\n open var align: Short\n open var meetOrSlice: Short\n\n companion object {\n val
SVG_PRESERVEASPECTRATIO_UNKNOWN: Short\n val SVG_PRESERVEASPECTRATIO_NONE:
Short\n val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short\n val
SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short\n val
SVG_PRESERVEASPECTRATIO_XMINYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMIDYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMINYMAX: Short\n val
SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short\n val SVG_MEETORSLICE_UNKNOWN: Short\n
val SVG_MEETORSLICE_MEET: Short\n val SVG_MEETORSLICE_SLICE: Short\n }\n}\n\n/**\n *
Exposes the JavaScript
[SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspect
Ratio) to Kotlin\n */\npublic external abstract class
SVGAnimatedPreserveAspectRatio {\n open val baseVal: SVGPreserveAspectRatio\n open val animVal:
SVGPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript
[SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin\n */\npublic
external abstract class SVGPathElement : SVGGeometryElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin\n */\npublic
external abstract class SVGRectElement : SVGGeometryElement {\n open val x: SVGAnimatedLength\n open
val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n
open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n companion object {\n val

```

```

ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n
 val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin\n
*\npublic external abstract class SVGCircleElement : SVGGeometryElement {\n open val cx:
SVGAnimatedLength\n open val cy: SVGAnimatedLength\n open val r: SVGAnimatedLength\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin\n
*\npublic external abstract class SVGEllipseElement : SVGGeometryElement {\n open val cx: SVGAnimatedLength\n
 open val cy: SVGAnimatedLength\n open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n
 val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin\n
*\npublic external abstract class SVGLineElement : SVGGeometryElement {\n open val x1: SVGAnimatedLength\n
 open val y1: SVGAnimatedLength\n open val x2: SVGAnimatedLength\n open val y2: SVGAnimatedLength\n\n
 companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin\n
*\npublic

```



```

external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin\n */\n\npublic external interface SVGAnimatedPoints {\n val points: SVGPointList\n val animatedPoints: SVGPointList\n}\n\npublic external abstract class SVGPointList {\n open val length: Int\n open val numberOfItems: Int\n fun clear(): DOMPoint\n fun initialize(newItem: DOMPoint): DOMPoint\n fun insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint\n fun replaceItem(newItem: DOMPoint, index: Int): DOMPoint\n fun removeItem(index: Int): DOMPoint\n fun appendItem(newItem: DOMPoint): DOMPoint\n fun getItem(index: Int): DOMPoint\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.get(index: Int): DOMPoint? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.set(index: Int, newItem: DOMPoint) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript [SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin\n */\n\npublic external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin\n */\n\npublic external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin\n */\n\npublic external abstract class SVGTextContentElement : SVGGraphicsElement {\n open val textLength:

```

```

SVGAnimatedLength\n open val lengthAdjust: SVGAnimatedEnumeration\n fun getNumberOfChars(): Int\n fun getComputedTextLength(): Float\n fun getSubStringLength(charnum: Int, nchars: Int): Float\n fun getStartPositionOfChar(charnum: Int): DOMPoint\n fun getEndPositionOfChar(charnum: Int): DOMPoint\n fun getExtentOfChar(charnum: Int): DOMRect\n fun getRotationOfChar(charnum: Int): Float\n fun getCharNumAtPosition(point: DOMPoint): Int\n fun selectSubString(charnum: Int, nchars: Int)\n\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n * Exposes the JavaScript [SVGTextPositioningElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPositioningElement) to Kotlin\n\n * public external abstract class SVGTextPositioningElement : SVGTextContentElement {\n open val x: SVGAnimatedLengthList\n open val y: SVGAnimatedLengthList\n open val dx: SVGAnimatedLengthList\n open val dy: SVGAnimatedLengthList\n open val rotate: SVGAnimatedNumberList\n\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n * Exposes the JavaScript [SVGTextElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextElement) to Kotlin\n\n * public external abstract class SVGTextElement : SVGTextPositioningElement {\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n\n * Exposes the JavaScript [SVGTSpanElement](https://developer.mozilla.org/en/docs/Web/API/SVGTSpanElement) to Kotlin\n\n * public external abstract class SVGTSpanElement : SVGTextPositioningElement {\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n
```

```

val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin\n
*/\npublic external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference {\n open
val startOffset: SVGAnimatedLength\n open val method: SVGAnimatedEnumeration\n open val spacing:
SVGAnimatedEnumeration\n\n companion object {\n val TEXTPATH_METHODTYPE_UNKNOWN:
Short\n val TEXTPATH_METHODTYPE_ALIGN: Short\n val
TEXTPATH_METHODTYPE_STRETCH: Short\n val TEXTPATH_SPACINGTYPE_UNKNOWN: Short\n
 val TEXTPATH_SPACINGTYPE_AUTO: Short\n val TEXTPATH_SPACINGTYPE_EXACT: Short\n
 val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin\n
*/\npublic external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference,
HTMLOrSVGImageElement {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open
val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio\n open var crossOrigin: String?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement)
to Kotlin\n
*/\npublic external abstract class SVGForeignObjectElement : SVGGraphicsElement {\n open val x:
SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val
height: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val

```

```

ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC:
Short\n }\n}\n\npublic external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {\n open
val refX: SVGAnimatedLength\n open val refY: SVGAnimatedLength\n open val markerUnits:
SVGAnimatedEnumeration\n open val markerWidth: SVGAnimatedLength\n open val markerHeight:
SVGAnimatedLength\n open val orientType: SVGAnimatedEnumeration\n open val orientAngle:
SVGAnimatedAngle\n open var orient: String\n fun setOrientToAuto()\n fun setOrientToAngle(angle:
SVGAngle)\n}\n companion object {\n val SVG_MARKERUNITS_UNKNOWN: Short\n val
SVG_MARKERUNITS_USERSPACEONUSE: Short\n val SVG_MARKERUNITS_STROKEWIDTH:
Short\n val SVG_MARKER_ORIENT_UNKNOWN: Short\n val SVG_MARKER_ORIENT_AUTO:
Short\n val SVG_MARKER_ORIENT_ANGLE: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n
 val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin\n
*\n\npublic external abstract class SVGSolidcolorElement : SVGElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n
 val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin\n
*\n\npublic external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {\n
 open val gradientUnits: SVGAnimatedEnumeration\n open val gradientTransform: SVGAnimatedTransformList\n
 open val spreadMethod: SVGAnimatedEnumeration\n}\n companion object {\n val
SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val
SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val
SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val

```

```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript
[SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to
Kotlin\n */\npublic external abstract class SVGLinearGradientElement : SVGGradientElement {\n open val x1:
SVGAnimatedLength\n open val y1: SVGAnimatedLength\n open val x2: SVGAnimatedLength\n open val
y2: SVGAnimatedLength\n\n companion object {\n val SVG_SPREADMETHOD_UNKNOWN: Short\n
 val SVG_SPREADMETHOD_PAD: Short\n val SVG_SPREADMETHOD_REFLECT: Short\n val
SVG_SPREADMETHOD_REPEAT: Short\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val
SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n
 val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
 val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n
 val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n
 val DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n
 val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
 }\n}\n}\n\n/**\n * Exposes the JavaScript
[SVGRadialGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGRadialGradientElement) to
Kotlin\n */\npublic external abstract class SVGRadialGradientElement : SVGGradientElement {\n open val cx:
SVGAnimatedLength\n open val cy: SVGAnimatedLength\n open val r: SVGAnimatedLength\n open val fx:
SVGAnimatedLength\n open val fy: SVGAnimatedLength\n open val fr: SVGAnimatedLength\n\n companion
object {\n val SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD:
Short\n val SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT:
Short\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE:
Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n
 val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n
 val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n
 val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
 val DOCUMENT_FRAGMENT_NODE: Short\n
 val NOTATION_NODE: Short\n
 val DOCUMENT_POSITION_DISCONNECTED: Short\n
 val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n
 val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
 }\n}\n}\n\npublic external
abstract class SVGMeshGradientElement : SVGGradientElement {\n companion object {\n val
SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val
SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val
SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n
 val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
 val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n
 val ENTITY_NODE: Short\n val

```

```

PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshrowElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshpatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin\n *\npublic
external abstract class SVGStopElement : SVGElement {\n open val offset: SVGAnimatedNumber\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n *
Exposes the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement)
to Kotlin\n *\npublic external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox,
SVGURIReference, SVGUnitTypes {\n open val patternUnits: SVGAnimatedEnumeration\n open val
patternContentUnits: SVGAnimatedEnumeration\n open val patternTransform: SVGAnimatedTransformList\n
open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n
open val height: SVGAnimatedLength\n\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN:
Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val

```

```

ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGHatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE:
Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external
abstract class SVGHatchpathElement : SVGElement {\n companion object {\n val ELEMENT_NODE:
Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE:
Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING:
Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val
DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin\n */\n\npublic
external abstract class SVGCursorElement : SVGElement, SVGURIReference {\n open val x:
SVGAnimatedLength\n open val y: SVGAnimatedLength\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:
Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val
DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the
JavaScript [SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin\n
*/\n\npublic external abstract class SVGScriptElement : SVGElement, SVGURIReference,
HTMLOrSVGScriptElement {\n open var type: String\n open var crossOrigin: String?\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

```

```

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE:
Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes
the JavaScript [SVGAElement](https://developer.mozilla.org/en/docs/Web/API/SVGAElement) to Kotlin\n
*\npublic external abstract class SVGAElement : SVGGraphicsElement, SVGURIReference {\n open val target:
SVGAnimatedString\n open val download: SVGAnimatedString\n open val rel: SVGAnimatedString\n open
val relList: SVGAnimatedString\n open val hreflang: SVGAnimatedString\n open val type:
SVGAnimatedString\n}\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin\n
*\npublic external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {\n companion
object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE:
Short\n val SVG_ZOOMANDPAN_MAGNIFY: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}"/>\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for
details\n\npackage org.w3c.files\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport
org.w3c.dom.events.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript
[Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin\n
*\npublic external open class
Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) :
MediaProvider, ImageBitmapSource {\n open val size: Number\n open val type: String\n open val isClosed:
Boolean\n fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String =
definedExternally): Blob\n fun close()\n}\n\npublic external interface BlobPropertyBag {\n var type: String? /*
= \"\" */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun BlobPropertyBag(type: String? = \"\"):
BlobPropertyBag {\n val o = js(\"({})\")\n

```



```

o["type"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript
[File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin\n */\npublic external open class File(fileBits:
Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {\n open val name:
String\n open val lastModified: Int\n}\n\npublic external interface FilePropertyBag : BlobPropertyBag {\n var
lastModified: Int?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun FilePropertyBag(lastModified: Int? =
undefined, type: String? = ""): FilePropertyBag {\n val o = js("{}")\n o["lastModified"] = lastModified\n
o["type"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript
[FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin\n */\npublic external abstract class
FileList : ItemArrayLike<File>
{\n override fun item(index: Int): File?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun FileList.get(index: Int): File?
= asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin\n */\npublic external open class
FileReader : EventTarget {\n open val readyState: Short\n open val result: dynamic\n open val error:
dynamic\n var onloadstart: ((ProgressEvent) -> dynamic)?\n var onprogress: ((ProgressEvent) -> dynamic)?\n
var onload: ((Event) -> dynamic)?\n var onabort: ((Event) -> dynamic)?\n var onerror: ((Event) ->
dynamic)?\n var onloadend: ((Event) -> dynamic)?\n fun readAsArrayBuffer(blob: Blob)\n fun readAsBinaryString(blob:
Blob)\n fun readAsText(blob: Blob, label: String = definedExternally)\n fun readAsDataURL(blob: Blob)\n
fun abort()\n\n companion object {\n val EMPTY: Short\n val LOADING: Short\n val DONE: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin\n */\npublic external
open class FileReaderSync {\n fun readAsArrayBuffer(blob: Blob): ArrayBuffer\n fun readAsBinaryString(blob:
Blob): String\n fun readAsText(blob: Blob, label: String = definedExternally): String\n fun
readAsDataURL(blob: Blob): String\n}\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n// See
github.com/kotlin/dukat for details\n\npackage org.w3c.notifications\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.events.*\nimport org.w3c.workers.*\n\n/**\n * Exposes the JavaScript
[Notification](https://developer.mozilla.org/en/docs/Web/API/Notification)
to Kotlin\n */\npublic external open class Notification(title: String, options: NotificationOptions =
definedExternally) : EventTarget {\n var onclick: ((MouseEvent) -> dynamic)?\n var onerror: ((Event) ->
dynamic)?\n open val title: String\n open val dir: NotificationDirection\n open val lang: String\n open val
body: String\n open val tag: String\n open val image: String\n open val icon: String\n open val badge:
String\n open val sound: String\n open val vibrate: Array<out Int>\n open val timestamp: Number\n open val
renotify: Boolean\n open val silent: Boolean\n open val noscreen: Boolean\n open val requireInteraction:
Boolean\n open val sticky: Boolean\n open val data: Any?\n open val actions: Array<out
NotificationAction>\n fun close()\n\n companion object {\n val permission: NotificationPermission\n
val maxActions: Int\n fun requestPermission(deprecatedCallback: (NotificationPermission)
-> Unit = definedExternally): Promise<NotificationPermission>\n }\n}\n\npublic external interface
NotificationOptions {\n var dir: NotificationDirection? /* = NotificationDirection.AUTO */\n get() =
definedExternally\n set(value) = definedExternally\n var lang: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var body: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var tag: String? /* = "" */\n get() =
definedExternally\n set(value) = definedExternally\n var image: String?\n get() = definedExternally\n
set(value) = definedExternally\n var icon: String?\n get() = definedExternally\n set(value) =
definedExternally\n var badge: String?\n get() = definedExternally\n set(value) = definedExternally\n
}

```

```

var sound: String? \n get() = definedExternally \n set(value) = definedExternally \n
 var vibrate: dynamic \n get() = definedExternally \n set(value) = definedExternally \n var timestamp:
Number? \n get() = definedExternally \n set(value) = definedExternally \n var renotify: Boolean? /* = false
/ \n get() = definedExternally \n set(value) = definedExternally \n var silent: Boolean? / = false */ \n
get() = definedExternally \n set(value) = definedExternally \n var noscreen: Boolean? /* = false */ \n get() =
definedExternally \n set(value) = definedExternally \n var requireInteraction: Boolean? /* = false */ \n get()
= definedExternally \n set(value) = definedExternally \n var sticky: Boolean? /* = false */ \n get() =
definedExternally \n set(value) = definedExternally \n var data: Any? /* = null */ \n get() =
definedExternally \n set(value) = definedExternally \n var actions: Array<NotificationAction>? /* = arrayOf()
*/ \n get() = definedExternally \n
 set(value) = definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n\n@kotlin.internal.InlineOnly \n\npublic inline fun NotificationOptions(dir:
NotificationDirection? = NotificationDirection.AUTO, lang: String? = "", body: String? = "", tag: String? = "",
image: String? = undefined, icon: String? = undefined, badge: String? = undefined, sound: String? = undefined,
vibrate: dynamic = undefined, timestamp: Number? = undefined, renotify: Boolean? = false, silent: Boolean? =
false, noscreen: Boolean? = false, requireInteraction: Boolean? = false, sticky: Boolean? = false, data: Any? = null,
actions: Array<NotificationAction>? = arrayOf()): NotificationOptions {\n val o = js("({})") \n o["dir"] = dir \n
 o["lang"] = lang \n o["body"] = body \n o["tag"] = tag \n o["image"] = image \n o["icon"] = icon \n
 o["badge"] = badge \n o["sound"] = sound \n o["vibrate"] = vibrate \n o["timestamp"] = timestamp \n
 o["renotify"] = renotify \n o["silent"] = silent \n o["noscreen"] = noscreen \n o["requireInteraction"] =
requireInteraction \n o["sticky"] = sticky \n o["data"] = data \n o["actions"] = actions \n return
o \n} \n\npublic external interface NotificationAction {\n var action: String? \n var title: String? \n var icon:
String? \n get() = definedExternally \n set(value) =
definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n\n@kotlin.internal.InlineOnly \n\npublic inline fun NotificationAction(action: String?,
title: String?, icon: String? = undefined): NotificationAction {\n val o = js("({})") \n o["action"] = action \n
 o["title"] = title \n o["icon"] = icon \n return o \n} \n\npublic external interface GetNotificationOptions {\n var
tag: String? /* = "" */ \n get() = definedExternally \n set(value) =
definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n\n@kotlin.internal.InlineOnly \n\npublic
inline fun GetNotificationOptions(tag: String? = ""): GetNotificationOptions {\n val o = js("({})") \n
 o["tag"] = tag \n return o \n} \n\n/** \n * Exposes the JavaScript
[NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin \n */ \n\npublic
external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) : ExtendableEvent {\n
 open val notification: Notification \n open val action: String \n\n companion object {\n val NONE: Short \n
 val CAPTURING_PHASE: Short \n val AT_TARGET: Short \n val BUBBLING_PHASE: Short \n
 } \n} \n\npublic external interface NotificationEventInit : ExtendableEventInit {\n var notification: Notification? \n
 var action: String? /* = "" */ \n get() = definedExternally \n set(value) =
definedExternally \n} \n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n\n@kotlin.internal.InlineOnly \n\npublic inline fun
NotificationEventInit(notification: Notification?, action: String? = "", bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): NotificationEventInit {\n val o = js("({})") \n o["notification"]
= notification \n o["action"] = action \n o["bubbles"] = bubbles \n o["cancelable"] = cancelable \n
 o["composed"] = composed \n return o \n} \n\n/* please, don't implement this interface!
*/ \n\n@JsName("null") \n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n\npublic external
interface NotificationPermission {\n companion object \n} \n\npublic inline val
NotificationPermission.Companion.DEFAULT: NotificationPermission get() =
"default".asDynamic().unsafeCast<NotificationPermission>() \n\npublic inline val

```

```

NotificationPermission.Companion.DENIED: NotificationPermission get() =
 \"denied\".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val
NotificationPermission.Companion.GRANTED: NotificationPermission get() =
 \"granted\".asDynamic().unsafeCast<NotificationPermission>()\n\n/*
 please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface NotificationDirection {\n companion object\n}\n\npublic inline val
NotificationDirection.Companion.AUTO: NotificationDirection get() =
 \"auto\".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.LTR: NotificationDirection get() =
 \"ltr\".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.RTL: NotificationDirection get() =
 \"rtl\".asDynamic().unsafeCast<NotificationDirection>()\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n//
See github.com/kotlin/dukat for details\n\npackage
org.w3c.workers\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport
org.w3c.dom.events.*\nimport org.w3c.fetch.*\nimport org.w3c.notifications.*\n\n/**\n * Exposes the JavaScript
[ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin\n */\n\npublic external
abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker,
UnionClientOrMessagePortOrServiceWorker {\n open val scriptURL: String\n open val state:
ServiceWorkerState\n open var onstatechange: ((Event) -> dynamic)?\n fun postMessage(message: Any?,
transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to
Kotlin\n */\n\npublic external abstract class ServiceWorkerRegistration : EventTarget {\n open val installing:
ServiceWorker?\n open val waiting: ServiceWorker?\n open val active: ServiceWorker?\n
 open val scope: String\n open var onupdatefound: ((Event) -> dynamic)?\n open val APISpace: dynamic\n
fun update(): Promise<Unit>\n fun unregister(): Promise<Boolean>\n fun showNotification(title: String,
options: NotificationOptions = definedExternally): Promise<Unit>\n fun getNotifications(filter:
GetNotificationOptions = definedExternally): Promise<Array<Notification>>\n fun methodName():
Promise<dynamic>\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin\n
*/\n\npublic external abstract class ServiceWorkerContainer : EventTarget {\n open val controller:
ServiceWorker?\n open val ready: Promise<ServiceWorkerRegistration>\n open var oncontrollerchange:
((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun register(scriptURL: String,
options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>\n fun
getRegistration(clientURL: String = definedExternally): Promise<Any?>\n fun getRegistrations():
Promise<Array<ServiceWorkerRegistration>>\n fun startMessages()\n}\n\npublic external interface
RegistrationOptions {\n var scope: String?\n get() = definedExternally\n set(value) = definedExternally\n
var type: WorkerType? /* = WorkerType.CLASSIC */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun RegistrationOptions(scope: String? =
undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {\n val o = js(\"{\}\")\n
o[\"scope\"] = scope\n o[\"type\"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to
Kotlin\n */\n\npublic external open class ServiceWorkerMessageEvent(type: String, eventInitDict:
ServiceWorkerMessageEventInit

```

```

= definedExternally) : Event {\n open val data: Any?\n open val origin: String\n open val lastEventId: String\n open val source: UnionMessagePortOrServiceWorker?\n open val ports: Array<out MessagePort>?\n\ncompanion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ServiceWorkerMessageEventInit : EventInit {\n var data: Any?\n get() = definedExternally\n set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String?\n get() = definedExternally\n set(value) = definedExternally\n var source: UnionMessagePortOrServiceWorker?\n get() = definedExternally\n set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ServiceWorkerMessageEventInit(data: Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source: UnionMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ServiceWorkerMessageEventInit {\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"origin\"] = origin\n o[\"lastEventId\"] = lastEventId\n o[\"source\"] = source\n o[\"ports\"] = ports\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to Kotlin\n *\npublic external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {\n open val clients: Clients\n\n open val registration: ServiceWorkerRegistration\n open var oninstall: ((Event) -> dynamic)?\n open var onactivate: ((Event) -> dynamic)?\n open var onfetch: ((FetchEvent) -> dynamic)?\n open var onforeignfetch: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n open var onnotificationclick: ((NotificationEvent) -> dynamic)?\n open var onnotificationclose: ((NotificationEvent) -> dynamic)?\n open var onfunctionalevent: ((Event) -> dynamic)?\n fun skipWaiting(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript [Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin\n *\npublic external abstract class Client : UnionClientOrMessagePortOrServiceWorker {\n open val url: String\n open val frameType: FrameType\n open val id: String\n fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [WindowClient](https://developer.mozilla.org/en/docs/Web/API/WindowClient) to Kotlin\n *\npublic external abstract class WindowClient : Client {\n open val visibilityState: dynamic\n open val focused: Boolean\n fun focus(): Promise<WindowClient>\n fun navigate(url: String): Promise<WindowClient>\n}\n\n/**\n * Exposes the JavaScript [Clients](https://developer.mozilla.org/en/docs/Web/API/Clients) to Kotlin\n *\npublic external abstract class Clients {\n fun get(id: String): Promise<Any?>\n fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>\n fun openWindow(url: String): Promise<WindowClient?>\n fun claim(): Promise<Unit>\n}\n\npublic external interface ClientQueryOptions {\n var includeUncontrolled: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var type: ClientType? /* = ClientType.WINDOW */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW): ClientQueryOptions {\n val o = js(\"({})\")\n o[\"includeUncontrolled\"] = includeUncontrolled\n o[\"type\"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript [ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin\n *\npublic external open class ExtendableEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : Event {\n fun waitUntil(f: Promise<Any?>)\n\ncompanion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ExtendableEventInit : EventInit\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```



```

String\n open val source: UnionClientOrMessagePortOrServiceWorker?\n open val ports: Array<out
MessagePort>?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n\n\npublic external interface
ExtendableMessageEventInit : ExtendableEventInit {\n var data: Any?\n get() =
definedExternally\n set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n
set(value) = definedExternally\n var lastEventId: String?\n get() = definedExternally\n set(value) =
definedExternally\n var source: UnionClientOrMessagePortOrServiceWorker?\n get() = definedExternally\n
 set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun ExtendableMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionClientOrMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableMessageEventInit {\n
 val o = js(\"({})\")\n o[\"data\"] = data\n
 o[\"origin\"] = origin\n o[\"lastEventId\"] = lastEventId\n o[\"source\"] = source\n o[\"ports\"] = ports\n
 o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return
o\n}\n\n\n/**\n * Exposes the JavaScript [Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin\n
*/\n\npublic external abstract class Cache {\n fun match(request: dynamic, options: CacheQueryOptions =
definedExternally): Promise<Any?>\n fun matchAll(request: dynamic = definedExternally, options:
CacheQueryOptions = definedExternally): Promise<Array<Response>>\n fun add(request: dynamic):
Promise<Unit>\n fun addAll(requests: Array<dynamic>): Promise<Unit>\n fun put(request: dynamic, response:
Response): Promise<Unit>\n fun delete(request: dynamic, options: CacheQueryOptions = definedExternally):
Promise<Boolean>\n fun keys(request: dynamic = definedExternally, options: CacheQueryOptions =
definedExternally): Promise<Array<Request>>\n}\n\n\npublic
external interface CacheQueryOptions {\n var ignoreSearch: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var ignoreMethod: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var ignoreVary: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var cacheName: String?\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun CacheQueryOptions(ignoreSearch:
Boolean? = false, ignoreMethod: Boolean? = false, ignoreVary: Boolean? = false, cacheName: String? = undefined):
CacheQueryOptions {\n val o = js(\"({})\")\n o[\"ignoreSearch\"] = ignoreSearch\n o[\"ignoreMethod\"] =
ignoreMethod\n o[\"ignoreVary\"] = ignoreVary\n o[\"cacheName\"] = cacheName\n return o\n}\n\n\npublic
external
interface CacheBatchOperation {\n var type: String?\n get() = definedExternally\n set(value) =
definedExternally\n var request: Request?\n get() = definedExternally\n set(value) = definedExternally\n
 var response: Response?\n get() = definedExternally\n set(value) = definedExternally\n var options:
CacheQueryOptions?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun CacheBatchOperation(type: String? =
undefined, request: Request? = undefined, response: Response? = undefined, options: CacheQueryOptions? =
undefined): CacheBatchOperation {\n val o = js(\"({})\")\n o[\"type\"] = type\n o[\"request\"] = request\n
 o[\"response\"] = response\n o[\"options\"] = options\n return o\n}\n\n\n\n/**\n * Exposes the JavaScript
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage)
to Kotlin\n */\n\npublic external abstract class CacheStorage {\n fun match(request: dynamic, options:
CacheQueryOptions = definedExternally): Promise<Any?>\n fun has(cacheName: String): Promise<Boolean>\n
 fun open(cacheName: String): Promise<Cache>\n fun delete(cacheName: String): Promise<Boolean>\n fun
keys(): Promise<Array<String>>\n}\n\n\npublic external open class FunctionalEvent : ExtendableEvent {\n

```







`@ExperimentalStdlibApi` must be accepted either by `@OptIn` annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`, or by using the compiler argument `-opt-in=kotlin.ExperimentalStdlibApi`.

`@RequiresOptIn(level = RequiresOptIn.Level.ERROR)` `@Retention(AnnotationRetention.BINARY)` `@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)` `@MustBeDocumented` `@SinceKotlin("1.3")` `public annotation class ExperimentalStdlibApi`

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

`@nimport kotlin.annotation.AnnotationTarget` `nimport kotlin.experimental.ExperimentalTypeInference`

`Allows to infer generic type arguments of a function from the calls in the annotated function parameter of that function. When this annotation is placed on a generic function parameter of a function, it enables to infer the type arguments of that generic function from the lambda body passed to that parameter. The calls that affect inference are either members of the receiver type of an annotated function parameter or extensions for that type. The extensions must be themselves annotated with @BuilderInference. Example: we declare fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>().-> Unit): Sequence<T> and use it like val result = sequence { yield("result") } Here the type argument of the resulting sequence is inferred to String from the argument of the [SequenceScope.yield] function, that is called inside the lambda passed to [sequence]. Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.`

`@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)` `@Retention(AnnotationRetention.BINARY)` `@SinceKotlin("1.3")` `@ExperimentalTypeInference` `public annotation class BuilderInference`

`Enables overload selection based on the type of the value returned from lambda argument. When two or more function overloads have otherwise the same parameter lists that differ only in the return type of a functional parameter, this annotation enables overload selection by the type of the value returned from the lambda function passed to this functional parameter. Example: fun create(intProducer: () -> Int): Int fun create(doubleProducer: () -> Double): Double val newValue = create { 3.14 }`

`The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned from the lambda function. This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported. Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.`

`@Target(FUNCTION)` `@Retention(AnnotationRetention.BINARY)` `@SinceKotlin("1.4")` `@ExperimentalTypeInference` `public annotation class OverloadResolutionByLambdaReturnType`

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

`nimport kotlin.annotation.AnnotationTarget` `nimport kotlin.internal.RequireKotlin` `nimport kotlin.internal.RequireKotlinVersionKind`

`The experimental multiplatform support API marker. Any usage of a declaration annotated with @ExperimentalMultiplatform must be accepted either by @OptIn annotation, e.g. @OptIn(ExperimentalMultiplatform::class), or by using the compiler argument -opt-in=kotlin.ExperimentalMultiplatform. @RequiresOptIn @MustBeDocumented @Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS) @Retention(AnnotationRetention.BINARY) public annotation class ExperimentalMultiplatform. Marks an expected annotation class that it isn't required to have actual counterparts in all platforms. This annotation is only applicable to expect annotation classes in multi-`

platform projects

and marks that class as `optional`. Optional expected class is allowed to have no corresponding actual class on the platform. Optional annotations can only be used to annotate something, not as types in signatures. If an optional annotation has no corresponding actual class on a platform, the annotation entries where it's used are simply erased when compiling code on that platform. Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.

```
/*\n@Target(ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@ExperimentalMultiplatform\npublic annotation class OptionalExpectation\n"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin\nProgramming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be\nfound in the license/LICENSE.txt file.\n *\npackage kotlin\nimport\nkotlin.annotation.AnnotationRetention.*\nimport kotlin.reflect.KClass\n/****\n
```

\* Signals that the annotated annotation class is a marker of an API that requires an explicit opt-in. Call sites of any declaration that is either annotated with such a marker or mentions in its signature any other declaration that requires opt-in should opt in to the API either by using `[OptIn]`, or by being annotated with that marker themselves, effectively causing further propagation of the opt-in requirement. The intended uses of opt-in markers include, but are not limited to the following:

- Experimental API for public preview that might change its semantics or affect binary compatibility.
- Internal declarations that should not be used outside the declaring library, but are `public` for technical reasons.
- Fragile or delicate API that needs a lot of expertise to use and thus require an explicit opt-in.

### Contagiousness

\* When a declaration is marked with an opt-in requirement, it is considered to be contagious,

meaning that all its uses or mentions in other declarations will require an explicit opt-in. A rule of thumb for propagating is the following: if the marked declaration ceases to exist, only the places with explicit opt-in (or the corresponding warning) will break. This rule does not imply transitivity, e.g. the propagation does not propagate opt-in through inlining, making it the responsibility of the `inline` function author to mark it properly.

### Type scopes

\* A type is considered requiring opt-in if it is marked with an opt-in marker, or the outer declaration (class or interface) requires opt-in. Any use of any declaration that mentions such type in its signature will require an explicit opt-in, even if it is not used directly on the call site, and even if such declarations do not require opt-in directly.

\* For example, consider the following declarations that are marked with non-propagating opt-in:

```
Unstable\n * @OptIn(UnstableApi::class)\n * fun foo(): Unstable = Unstable()\n *\n@OptIn(UnstableApi::class)\n * fun bar(arg: Unstable = Unstable()) {\n * }\n *\n@OptIn(UnstableApi::class)\n * fun\nUnstable?.baz() {\n * }\n *\n * and their respective call sites:\n * fun outerFun() {\n * val s = foo()\n * bar()\n * null.baz()\n * }\n *\n * Even though call sites do not mention `Unstable` type directly, the
```

corresponding opt-in warning or error will be triggered in each call site due to propagation contagiousness. Note that the propagation is not transitive, i.e. calls to `outerFun` itself would not trigger any further opt-in requirements.

### Lexical scopes

\* If a type requires an opt-in, such requirement is propagated to its lexical scope and all its nested declarations. For example, for the following scope:

```
Unstable\n * class Unstable {\n * fun memberFun() = ... \n * class NestedClass {\n * fun nestedFun()\n * }\n * }\n *\n * Any use of `Unstable`, `NestedClass`, or their member functions will require an
```

explicit opt-in.

### Overridden declarations

\* Opt-in markers are also propagated through the inheritance and interface implementation. If the base declaration requires an opt-in, overriding it requires either an explicit opt-in or propagating the opt-in requirement. See also [Kotlin language documentation](<https://kotlinlang.org/docs/opt-in-requirements.html>) for more information.

\* @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.

The default message is: `"This declaration is experimental and its usage should be marked with 'Marker' or '@OptIn(Marker::class)'"`, where `Marker` is the opt-in requirement marker.

\* @property level specifies how usages of API without an explicit opt-in are reported in code.

```

*/\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.3")\npublic annotation class
RequiresOptIn(\n val message: String = "",\n val level: Level = Level.ERROR)\n {\n /**\n * Severity of
the diagnostic that should be reported on usages which did not explicitly opt into
 * the API either by using
[OptIn] or by being annotated with the corresponding marker annotation.\n */\n public enum class Level {\n
/** Specifies that a warning should be reported on incorrect usages of this API. */\n WARNING,\n
/** Specifies that a compilation error should be reported on incorrect usages of this API. */\n ERROR,\n
}\n}\n\n/**\n * Allows to use the API denoted by the given markers in the annotated file, declaration, or
expression.\n * If a declaration is annotated with [OptIn], its usages are not required to opt in to that API.\n *
* [markerClass] specifies marker annotations that require explicit opt-in. The marker annotation
is\n * not required to be itself marked with [RequiresOptIn] to enable gradual migration of API from requiring opt-
in to the regular one,\n * yet declaring such `OptIn` yields a compilation warning.\n *
* See also [Kotlin language documentation](https://kotlinlang.org/docs/opt-in-requirements.html) for more information.\n *
* @property
markerClass specifies marker annotations that require explicit opt-in.\n * @see RequiresOptIn for a detailed
description of opt-in semantics and propagation rules.\n */\n@Target(\n CLASS, PROPERTY,
LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER,
PROPERTY_SETTER, EXPRESSION, FILE,
TYPEALIAS)\n\n@Retention(SOURCE)\n@SinceKotlin("1.3")\npublic annotation class OptIn(\n vararg val
markerClass: KClass<out Annotation>)\n\n/**\n * This annotation marks the experimental preview of the
language feature [SubclassOptInRequired].\n *
* > Note that this API is in a preview state and has a chance of
being changed in the
future.\n * Do not use it if you develop a library since your library can become source incompatible\n * with the
future versions of Kotlin.\n */\n@Target(CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.8")\npublic
annotation class ExperimentalSubclassOptIn\n\n/**\n * Annotation that marks open for subclassing classes and
interfaces, and makes implementation\n * and extension of such declarations as requiring an explicit opt-in.\n *
\n * When applied, any attempt to subclass the target declaration will trigger an opt-in\n * with the corresponding level
and message.\n *
\n * The intended uses of subclass opt-in markers include, but are not limited to the following
API:\n * - Stable to use, but unstable to implement due to its further evolution.\n * - Stable to use, but closed for
3rd-part implementations due to internal or technical reasons.\n * - Stable to use, but delicate or fragile to
implement.\n * - Stable to use, but with a contract that may be weakened in the future in a backwards-
incompatible\n
* manner for external implementations.\n *
\n * Contrary to regular [RequiresOptIn], there are three ways to opt-in
into the subclassing requirement:\n * - Annotate declaration with the marker annotation, making it propagating.\n * -
Annotate declaration with [OptIn] in order to opt in into the provided guarantees in a non-propagating manner.\n * -
Annotate declaration with [SubclassOptInRequired] with the same marker class, making it further propagating only
for subclassing.\n *
\n * Uses of this annotation are limited to open and abstract classes, and non-`fun` interfaces.\n *
\n * Any other uses allowed by `CLASS` annotation target yield a compilation error.\n *
\n * @property markerClass
specifies marker annotation that require explicit opt-in.\n * @see RequiresOptIn for a detailed description of opt-in
semantics and propagation rules.\n
*/\n@Target(CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.8")\n@ExperimentalSubclassOptIn\npublic
annotation class SubclassOptInRequired(\n
 val markerClass: KClass<out Annotation>)\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport
kotlin.annotation.AnnotationRetention.BINARY\nimport kotlin.annotation.AnnotationTarget.*\nimport
kotlin.reflect.KClass\n\n\n@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
TYPEALIAS)\n@Retention(BINARY)\ninternal annotation class WasExperimental(\n vararg val markerClass:
KClass<out Annotation>)\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

```



```

_size)\n\n return list[fromIndex + index]\n }\n\n override val size: Int get() = _size\n }\n\n /**\n * Compares this list with other list instance with the ordered structural equality.\n * @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same order.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is List<*>) return false\n return orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this list.\n */\n override fun hashCode(): Int = orderedHashCode(this)\n\n private open inner class IteratorImpl : Iterator<E> {\n /**\n * the index of the item that will be returned on the next call to [next]()\n */\n protected var index = 0\n\n override fun hasNext(): Boolean = index < size\n\n override fun next(): E {\n if (!hasNext())\n throw NoSuchElementException()\n return get(index++)\n }\n }\n\n /**\n * Implementation of [ListIterator] for abstract lists.\n */\n private open inner class ListIteratorImpl(index: Int) : IteratorImpl(),\n ListIterator<E> {\n init {\n checkPositionIndex(index, this@AbstractList.size)\n this.index = index\n }\n\n override fun hasPrevious(): Boolean = index > 0\n\n override fun nextIndex(): Int = index\n\n override fun previous(): E {\n if (!hasPrevious()) throw NoSuchElementException()\n return get(--index)\n }\n\n override fun previousIndex(): Int = index - 1\n }\n\n internal companion object {\n internal fun checkElementIndex(index: Int, size: Int) {\n if (index < 0 || index >= size) {\n throw IndexOutOfBoundsException("index: $index, size: $size")\n }\n }\n\n internal fun checkPositionIndex(index:\n Int, size: Int) {\n if (index < 0 || index > size) {\n throw IndexOutOfBoundsException("index: $index, size: $size")\n }\n }\n\n internal fun checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {\n if (fromIndex < 0 || toIndex > size) {\n throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex, size: $size")\n }\n if (fromIndex > toIndex) {\n throw IllegalArgumentException("fromIndex: $fromIndex > toIndex: $toIndex")\n }\n }\n\n internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {\n if (startIndex < 0 || endIndex > size) {\n throw IndexOutOfBoundsException("startIndex: $startIndex, endIndex: $endIndex, size: $size")\n }\n if (startIndex > endIndex) {\n throw IllegalArgumentException("startIndex: $startIndex > endIndex: $endIndex")\n }\n }\n\n internal fun orderedHashCode(c: Collection<*>): Int {\n var hashCode = 1\n for (e in c) {\n hashCode = 31 * hashCode + (e?.hashCode() ?: 0)\n }\n return hashCode\n }\n\n internal fun orderedEquals(c: Collection<*>, other: Collection<*>): Boolean {\n if (c.size != other.size) return false\n val otherIterator = other.iterator()\n for (elem in c) {\n val elemOther = otherIterator.next()\n if (elem != elemOther) {\n return false\n }\n }\n return true\n }\n }\n}\n\n"/>\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n * Based on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\npackage\n\nkotlin.collections\n\n/**\n * Provides a skeletal implementation of the read-only [Map] interface.\n */\n * The implementor is required to implement [entries] property, which should return read-only set of map entries.\n */\n * @param K the type of map keys. The map is invariant in its key type.\n * @param V the type of map values. The map is covariant in its value type.\n */\n * @SinceKotlin("1.1")\npublic abstract class AbstractMap<K, out V>\nprotected constructor() : Map<K, V> {\n override fun containsKey(key: K): Boolean {\n return implFindEntry(key) != null\n }\n\n override fun containsValue(value: @UnsafeVariance V): Boolean = entries.any { it.value == value }\n\n internal fun containsEntry(entry: Map.Entry<*, *>): Boolean {\n // since entry comes from @UnsafeVariance parameters it can be virtually anything\n if (entry !is Map.Entry<*, *>) return false\n val key = entry.key\n val value = entry.value\n val ourValue = get(key)\n if (value != ourValue) {\n return false\n }\n // Perhaps it was null and we don't contain the key?\n if (ourValue == null && !containsKey(key)) {\n return false\n }\n return true\n }\n}\n\n/**\n * Compares this map with other instance with the ordered structural equality.\n * @return true, if [other] instance is a [Map] of the same size, all entries of which are contained in the [entries] set of this

```

```

map.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is Map<*, *>) return false\n if (size != other.size) return false\n return other.entries.all { containsEntry(it) }\n }\n\n override operator fun get(key: K): V? = implFindEntry(key)?.value\n\n /**\n * Returns the hash code value for this map.\n * It is the same as the hashCode of [entries] set.\n */\n override fun hashCode(): Int = entries.hashCode()\n\n override fun isEmpty(): Boolean = size == 0\n\n override val size: Int get() = entries.size\n\n /**\n * Returns a read-only [Set] of all keys in this map.\n * Accessing this property first time creates a keys view from [entries].\n * All subsequent accesses just return the created instance.\n */\n override val keys: Set<K>\n get() {\n if (_keys == null) {\n _keys = object : AbstractSet<K>()\n }\n override operator fun contains(element: K): Boolean = containsKey(element)\n }\n\n override operator fun iterator(): Iterator<K> {\n val entryIterator = entries.iterator()\n return object : Iterator<K> {\n override fun hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): K = entryIterator.next().key\n }\n }\n\n override val size: Int get() = this@AbstractMap.size\n\n @kotlin.jvm.Volatile\n private var _keys: Set<K>? = null\n\n override fun toString(): String = entries.joinToString(", ", "\n", "\n") { toString(it) }\n\n private fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)\n\n private fun toString(o: Any?): String = if (o === this) \"(this Map)\" else o.toString()\n\n /**\n * Returns a read-only [Collection] of all values in this map.\n * Accessing this property first time creates a values view from [entries].\n * All subsequent accesses just return the created instance.\n */\n override val values: Collection<V>\n get() {\n if (_values == null) {\n _values = object : AbstractCollection<V>()\n }\n override operator fun contains(element: V): Boolean = containsValue(element)\n }\n\n override operator fun iterator(): Iterator<V> {\n val entryIterator = entries.iterator()\n return object : Iterator<V> {\n override fun hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): V = entryIterator.next().value\n }\n }\n\n override val size: Int get() = this@AbstractMap.size\n\n @kotlin.jvm.Volatile\n private var _values: Collection<V>? = null\n\n private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }\n\n internal companion object {\n internal fun entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?: 0) xor (value?.hashCode() ?: 0) }\n internal fun entryToString(e: Map.Entry<*, *>): String = with(e) { \"$key=$value\" }\n internal fun entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {\n if (other !is Map.Entry<*, *>) return false\n return e.key == other.key && e.value == other.value\n }\n }\n\n /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n /**\n * Provides a skeletal implementation of the read-only [Set] interface.\n * This class is intended to help implementing read-only sets so it doesn't support concurrent modification tracking.\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n */\n @SinceKotlin("1.1")\n public abstract class AbstractSet<out E> protected constructor() : AbstractCollection<E>(), Set<E> {\n /**\n * Compares this set with other set instance with the unordered structural equality.\n * @return true, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is Set<*>) return false\n return setEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this set.\n */\n override fun hashCode(): Int = unorderedHashCode(this)\n\n internal companion object {\n internal fun unorderedHashCode(c: Collection<*>): Int {\n var hashCode = 0\n for (element in c) {\n hashCode += (element?.hashCode() ?: 0)\n }\n return hashCode\n }\n\n internal fun setEquals(c: Set<*>, other: Set<*>): Boolean {\n if (c.size != other.size) return false\n return c.containsAll(other)\n }\n }\n }\n\n /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin

```

Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*^\n\npackage kotlin.collections\n\n/\*\*\n \* Resizable-array implementation of the deque data structure.\n \*^\n \* The name deque is short for \"double ended queue\" and is usually pronounced \"deck\".\n \*^\n \* The collection provide methods for convenient access to the both ends.\n \*^\n \* It also implements [MutableList] interface and supports efficient get/set operations by index.\n

```

*^\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class ArrayDeque<E> :
AbstractMutableList<E> {\n private var head: Int = 0\n private var elementData: Array<Any?>\n\n override
var size: Int = 0\n private set\n\n /**\n * Constructs an empty deque with specified [initialCapacity], or
throws [IllegalArgumentException] if [initialCapacity] is negative.\n *^\n public constructor(initialCapacity:
Int) {\n
 elementData = when {\n initialCapacity == 0 -> emptyElementData\n initialCapacity > 0 ->
arrayOfNulls(initialCapacity)\n else -> throw IllegalArgumentException(\"Illegal Capacity:
${initialCapacity}\")\n }\n }\n\n /**\n * Constructs an empty deque.\n *^\n public constructor() {\n
elementData = emptyElementData\n }\n\n /**\n * Constructs a deque that contains the same elements as the
specified [elements] collection in the same order.\n *^\n public constructor(elements: Collection<E>) {\n
elementData = elements.toArray()\n size = elementData.size\n if (elementData.isEmpty())
elementData = emptyElementData\n }\n\n /**\n * Ensures that the capacity of this deque is at least equal to
the specified [minCapacity].\n *^\n * If the current capacity is less than the [minCapacity], a new backing
storage is allocated with greater capacity.\n *^\n * Otherwise, this method takes
no action and simply returns.\n *^\n private fun ensureCapacity(minCapacity: Int) {\n if (minCapacity < 0)
throw IllegalStateException(\"Deque is too big.\") // overflow\n if (minCapacity <= elementData.size) return\n
 if (elementData === emptyElementData) {\n elementData =
arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))\n return\n }\n val newCapacity =
newCapacity(elementData.size, minCapacity)\n copyElements(newCapacity)\n }\n\n /**\n * Creates a
new array with the specified [newCapacity] size and copies elements in the [elementData] array to it.\n *^\n private fun copyElements(newCapacity: Int) {\n val newElements = arrayOfNulls<Any?>(newCapacity)\n
elementData.copyInto(newElements, 0, head, elementData.size)\n elementData.copyInto(newElements,
elementData.size - head, 0, head)\n head = 0\n elementData = newElements\n }\n\n @kotlin.internal.InlineOnly\n
private inline fun internalGet(internalIndex: Int): E {\n @Suppress(\"UNCHECKED_CAST\")\n return
elementData[internalIndex] as E\n }\n\n private fun positiveMod(index: Int): Int = if (index >=
elementData.size) index - elementData.size else index\n\n private fun negativeMod(index: Int): Int = if (index < 0)
index + elementData.size else index\n\n @kotlin.internal.InlineOnly\n private inline fun internalIndex(index:
Int): Int = positiveMod(head + index)\n\n private fun incremented(index: Int): Int = if (index ==
elementData.lastIndex) 0 else index + 1\n\n private fun decremented(index: Int): Int = if (index == 0)
elementData.lastIndex else index - 1\n\n override fun isEmpty(): Boolean = size == 0\n\n /**\n * Returns the
first element, or throws [NoSuchElementException] if this deque is empty.\n *^\n public fun first(): E = if
(isEmpty()) throw NoSuchElementException(\"ArrayDeque is empty.\") else internalGet(head)\n\n /**\n * Returns the first element, or `null` if this deque is empty.\n *^\n public fun firstOrNull(): E? = if
(isEmpty()) null else internalGet(head)\n\n /**\n * Returns the last element, or throws
[NoSuchElementException] if this deque is empty.\n *^\n public fun last(): E = if (isEmpty()) throw
NoSuchElementException(\"ArrayDeque is empty.\") else internalGet(internalIndex(lastIndex))\n\n /**\n *
Returns the last element, or `null` if this deque is empty.\n *^\n public fun lastOrNull(): E? = if (isEmpty()) null
else internalGet(internalIndex(lastIndex))\n\n /**\n * Prepends the specified [element] to this deque.\n *^\n
public fun addFirst(element: E) {\n ensureCapacity(size + 1)\n head = decremented(head)\n
elementData[head] = element\n size += 1\n }\n\n /**\n * Appends the specified [element] to this deque.\n
 *^\n public fun addLast(element: E) {\n ensureCapacity(size +

```

```

1)\n\n elementData[internalIndex(size)] = element\n size += 1\n }\n\n /**\n * Removes the first
element from this deque and returns that removed element, or throws [NoSuchElementException] if this deque is
empty.\n */\n public fun removeFirst(): E {\n if (isEmpty()) throw NoSuchElementException("ArrayDeque
is empty.")\n val element = internalGet(head)\n elementData[head] = null\n head =
incremented(head)\n size -= 1\n return element\n }\n\n /**\n * Removes the first element from this
deque and returns that removed element, or returns `null` if this deque is empty.\n */\n public fun
removeFirstOrNull(): E? = if (isEmpty()) null else removeFirst()\n\n /**\n * Removes the last element from this
deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty.\n */\n
public fun removeLast(): E {\n if (isEmpty()) throw NoSuchElementException("ArrayDeque
is empty.")\n val internalLastIndex = internalIndex(lastIndex)\n val element =
internalGet(internalLastIndex)\n elementData[internalLastIndex] = null\n size -= 1\n return element\n
}\n\n /**\n * Removes the last element from this deque and returns that removed element, or returns `null` if
this deque is empty.\n */\n public fun removeLastOrNull(): E? = if (isEmpty()) null else removeLast()\n\n //
MutableList, MutableCollection\n public override fun add(element: E): Boolean {\n addLast(element)\n
return true\n }\n\n public override fun add(index: Int, element: E) {\n
AbstractList.checkPositionIndex(index, size)\n if (index == size) {\n addLast(element)\n
return\n } else if (index == 0) {\n addFirst(element)\n return\n }\n ensureCapacity(size
+ 1)\n // Elements in circular array lay in 2 ways:\n //
1. `head` is less than `tail`: [#, #, e1, e2, e3, #]\n // 2. `head` is greater than `tail`: [e3, #, #, #, e1, e2]\n
// where head is the index of the first element in the circular array,\n // and tail is the index following the last
element.\n //\n // At this point the insertion index is not equal to head or tail.\n // Also the circular array
can store at least one more element.\n //\n // Depending on where the given element must be inserted the
preceding or the succeeding\n // elements will be shifted to make room for the element to be inserted.\n //\n
// In case the preceding elements are shifted:\n // * if the insertion index is greater than the head (regardless
of circular array form)\n // -> shift the preceding elements\n // * otherwise, the circular array has (2)
form and the insertion index is less than tail\n // -> shift all elements in the back of
the array\n // -> shift preceding elements in the front of the array\n // In case the succeeding elements
are shifted:\n // * if the insertion index is less than the tail (regardless of circular array form)\n // ->
shift the succeeding elements\n // * otherwise, the circular array has (2) form and the insertion index is greater
than head\n // -> shift all elements in the front of the array\n // -> shift succeeding elements in the
back of the array\n val internalIndex = internalIndex(index)\n if (index < (size + 1) shr 1) {\n //
closer to the first element -> shift preceding elements\n val decrementedInternalIndex =
decremented(internalIndex)\n val decrementedHead = decremented(head)\n if
(decrementedInternalIndex >= head) {\n elementData[decrementedHead] = elementData[head] // head can
be zero\n elementData.copyInto(elementData,
head, head + 1, decrementedInternalIndex + 1)\n } else { // head > tail\n
elementData.copyInto(elementData, head - 1, head, elementData.size) // head can't be zero\n
elementData[elementData.size - 1] = elementData[0]\n elementData.copyInto(elementData, 0, 1,
decrementedInternalIndex + 1)\n }\n elementData[decrementedInternalIndex] = element\n
head = decrementedHead\n } else {\n // closer to the last element -> shift succeeding elements\n
val tail = internalIndex(size)\n if (internalIndex < tail) {\n elementData.copyInto(elementData,
internalIndex + 1, internalIndex, tail)\n } else { // head > tail\n elementData.copyInto(elementData,
1, 0, tail)\n elementData[0] = elementData[elementData.size - 1]\n
elementData.copyInto(elementData, internalIndex + 1, internalIndex, elementData.size
- 1)\n }\n elementData[internalIndex] = element\n }\n size += 1\n }\n\n private fun
copyCollectionElements(internalIndex: Int, elements: Collection<E>) {\n val iterator = elements.iterator()\n
for (index in internalIndex until elementData.size) {\n if (!iterator.hasNext()) break\n
elementData[index] = iterator.next()\n }\n for (index in 0 until head) {\n if (!iterator.hasNext())

```





```

(index in 0 until tail) {\n if (element == elementData[index]) return index + elementData.size - head\n
 }\n }\n return -1\n }\n public override fun lastIndexOf(element: E): Int {\n val tail =\ninternalIndex(size)\n\n if (head < tail) {\n for (index in tail - 1 downTo head) {\n if (element == elementData[index])\nreturn index - head\n }\n } else if (head > tail) {\n for (index in tail - 1 downTo 0) {\n if\n(element == elementData[index]) return index + elementData.size - head\n }\n } for (index in\n elementData.lastIndex downTo head) {\n if (element == elementData[index]) return index - head\n }\n }\n return -1\n }\n public override fun remove(element: E): Boolean {\n val index =\n indexOf(element)\n if (index == -1) return false\n removeAt(index)\n return true\n }\n\n public\n override fun removeAt(index: Int): E {\n AbstractList.checkNotNull(index, size)\n if (index ==\n lastIndex) {\n return removeLast()\n } else if (index == 0) {\n return removeFirst()\n }\n val internalIndex = internalIndex(index)\n val element = internalGet(internalIndex)\n if\n(index < size shr 1) {\n // closer to the first element -> shift preceding elements\n if (internalIndex >=\n head) {\n elementData.copyInto(elementData, head + 1, head, internalIndex)\n } else { // head >\ntail, internalIndex < head\n elementData.copyInto(elementData, 1, 0, internalIndex)\n }\n elementData[0] = elementData[elementData.size - 1]\n elementData.copyInto(elementData, head + 1,\n head, elementData.size - 1)\n }\n elementData[head] = null\n head = incremented(head)\n } else {\n // closer to the last element -> shift succeeding elements\n val internalLastIndex =\n internalIndex(lastIndex)\n if (internalIndex <= internalLastIndex) {\n elementData.copyInto(elementData, internalIndex, internalIndex\n + 1, internalLastIndex + 1)\n } else { // head > tail, internalIndex > head\n elementData.copyInto(elementData, internalIndex, internalIndex + 1, elementData.size)\n elementData[elementData.size - 1] = elementData[0]\n elementData.copyInto(elementData, 0, 1,\n internalLastIndex + 1)\n }\n elementData[internalLastIndex] = null\n }\n size -= 1\n }\n return element\n }\n\n public override fun removeAll(elements: Collection<E>): Boolean = filterInPlace {\n !elements.contains(it) }\n\n public override fun retainAll(elements: Collection<E>): Boolean = filterInPlace {\n elements.contains(it) }\n\n private inline fun filterInPlace(predicate: (E) -> Boolean): Boolean {\n if\n (this.isEmpty() || elementData.isEmpty())\n return false\n\n val tail = internalIndex(size)\n var\n newTail = head\n var modified = false\n\n if (head < tail) {\n for (index in head until tail) {\n val element = elementData[index]\n\n @Suppress("UNCHECKED_CAST")\n if (predicate(element as E))\n elementData[newTail++] = element\n else\n modified = true\n }\n\n elementData.fill(null, newTail, tail)\n } else {\n for (index in head until elementData.size) {\n val element = elementData[index]\n elementData[index] = null\n\n @Suppress("UNCHECKED_CAST")\n if (predicate(element as E))\n elementData[newTail++] = element\n else\n modified = true\n }\n\n newTail =\n positiveMod(newTail)\n\n for (index in 0 until tail) {\n val element = elementData[index]\n elementData[index] = null\n\n @Suppress("UNCHECKED_CAST")\n if (predicate(element as E)) {\n elementData[newTail] = element\n newTail =\n incremented(newTail)\n } else {\n modified = true\n }\n }\n\n if\n (modified)\n size = negativeMod(newTail - head)\n return modified\n }\n\n public override fun\n clear() {\n val tail = internalIndex(size)\n if (head < tail) {\n elementData.fill(null, head, tail)\n }\n else if (isNotEmpty()) {\n elementData.fill(null, head, elementData.size)\n elementData.fill(null, 0,\n tail)\n }\n head = 0\n size = 0\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override\n fun <T> toArray(array: Array<T>): Array<T> {\n @Suppress("UNCHECKED_CAST")\n val dest = (if\n (array.size >= size) array else arrayOfNulls(array, size)) as Array<Any?>\n val tail = internalIndex(size)\n if (head

```

```

< tail) {\n elementData.copyInto(dest, startIndex = head, endIndex = tail)\n } else if (isEmpty()) {\n elementData.copyInto(dest, destinationOffset = 0, startIndex = head, endIndex = elementData.size)\n }\n elementData.copyInto(dest, destinationOffset = elementData.size - head, startIndex = 0, endIndex = tail)\n }\n if (dest.size > size) {\n dest[size] = null // null-terminate\n }\n\n @Suppress("UNCHECKED_CAST")\n return dest as Array<T>\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override fun toArray(): Array<Any?> {\n return\n toArray(arrayOfNulls<Any?>(size))\n }\n\n // for testing\n internal fun <T> testToArray(array: Array<T>):\n Array<T> = toArray(array)\n internal fun testToArray(): Array<Any?> = toArray()\n\n internal companion\n object {\n private val emptyElementData = emptyArray<Any?>()\n private const val maxArraySize =\n Int.MAX_VALUE - 8\n\n private const val defaultMinCapacity = 10\n internal fun newCapacity(oldCapacity: Int, minCapacity: Int):\n Int {\n // overflow-conscious\n var newCapacity = oldCapacity + (oldCapacity shr 1)\n if\n (newCapacity - minCapacity < 0)\n newCapacity = minCapacity\n if (newCapacity - maxArraySize\n > 0)\n newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize\n }\n return newCapacity\n }\n }\n\n // For testing only\n internal fun internalStructure(structure: (head: Int,\n elements: Array<Any?>) -> Unit) {\n val tail = internalIndex(size)\n val head = if (isEmpty() || head < tail)\n head else head - elementData.size\n structure(head, toArray())\n }\n }"/*\n * Copyright 2010-2018 JetBrains\n s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0\n license that can be found in the license/LICENSE.txt\n file.\n */\n\n @file:kotlin.jvm.JvmMultifileClass\n @file:kotlin.jvm.JvmName("ArraysKt")\n\n \n package\n kotlin.collections\n\n import kotlin.contracts.*\n\n /**\n * Returns a single list of all elements from all arrays in the\n given array.\n * @sample samples.collections.Arrays.Transformations.flattenArray\n */\n\n public fun <T> Array<out\n Array<out T>>.flatten(): List<T> {\n val result = ArrayList<T>(sumOf { it.size })\n for (element in this) {\n result.addAll(element)\n }\n return result\n }\n\n /**\n * Returns a pair of lists, where\n * *first* list is built from\n the first values of each pair from this array,\n * *second* list is built from the second values of each pair from this\n array.\n * @sample samples.collections.Arrays.Transformations.unzipArray\n */\n\n public fun <T, R> Array<out\n Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val listT = ArrayList<T>(size)\n val listR =\n ArrayList<R>(size)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n }\n\n /**\n * Returns `true` if this nullable array is either null or empty.\n * @sample\n samples.collections.Arrays.Usage.arrayIsNullOrEmpty\n */\n\n @SinceKotlin("1.3")\n @kotlin.internal.InlineOnly\n public inline fun Array<*>.isNullOrEmpty(): Boolean\n {\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n return this == null ||\n this.isEmpty()\n }\n\n /**\n * Returns this array if it's not empty\n * or the result of calling [defaultValue] function if\n the array is empty.\n * @sample samples.collections.Arrays.Usage.arrayIfEmpty\n */\n\n @SinceKotlin("1.3")\n @kotlin.internal.InlineOnly\n @Suppress("UPPER_BOUND_CANNOT_BE_ARRAY")\n public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<*>, C : R =\n if (isEmpty())\n defaultValue() else\n this\n\n\n @OptIn(ExperimentalUnsignedTypes::class)\n @SinceKotlin("1.3")\n @PublishedApi\n @kotlin.jvm.Jvm\n Name("contentDeepEquals")\n @kotlin.js.JsName("contentDeepEqualsImpl")\n internal\n fun <T> Array<out T>?.contentDeepEqualsImpl(other: Array<out T>?): Boolean {\n if (this === other) return\n true\n if (this == null || other == null || this.size != other.size) return false\n for (i in indices) {\n val v1 =\n this[i]\n val v2 = other[i]\n if (v1 === v2) {\n continue\n } else if (v1 == null || v2 == null) {\n return false\n }\n when {\n v1 is Array<*> && v2 is Array<*> -> if\n (!v1.contentDeepEquals(v2)) return false\n v1 is ByteArray && v2 is ByteArray -> if\n (!v1.contentEquals(v2)) return false\n v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2))\n return false\n v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false\n v1 is\n LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false\n v1 is FloatArray && v2 is

```

```

FloatArray -> if (!v1.contentEquals(v2)) return false\n v1 is DoubleArray && v2 is DoubleArray -> if
(!v1.contentEquals(v2)) return false\n v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2))
return false\n v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n\n
v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false\n v1 is UShortArray
&& v2 is UShortArray -> if (!v1.contentEquals(v2)) return false\n v1 is UIntArray && v2 is UIntArray -
> if (!v1.contentEquals(v2)) return false\n v1 is ULongArray && v2 is ULongArray -> if
(!v1.contentEquals(v2)) return false\n\n else -> if (v1 != v2) return false\n }\n\n }\n return
true\n}\n\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.JvmName("contentDeepToString")\n@kotlin.js.
JsName("contentDeepToStringImpl")\ninternal fun <T> Array<out T>?.contentDeepToStringImpl():
String {\n if (this == null) return "null"\n val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2
// in order not to overflow Int.MAX_VALUE\n return buildString(length) {\n
contentDeepToStringInternal(this, mutableListOf())\n
}\n}\n\n@OptIn(ExperimentalUnsignedTypes::class)\nprivate fun <T> Array<out
T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {\n if (this in
processed) {\n result.append("[...]")\n return\n }\n processed.add(this)\n result.append("[")\n for (i
in indices) {\n if (i != 0) {\n result.append(", ")\n }\n val element = this[i]\n when
(element) {\n null -> result.append("null")\n is Array<*> ->
element.contentDeepToStringInternal(result, processed)\n is ByteArray ->
result.append(element.contentToString())\n is ShortArray -> result.append(element.contentToString())\n
 is IntArray -> result.append(element.contentToString())\n is LongArray ->
result.append(element.contentToString())\n is FloatArray -> result.append(element.contentToString())\n
 is DoubleArray -> result.append(element.contentToString())\n is CharArray ->
result.append(element.contentToString())\n is BooleanArray -> result.append(element.contentToString())\n
 is UByteArray -> result.append(element.contentToString())\n is UShortArray ->
result.append(element.contentToString())\n is UIntArray -> result.append(element.contentToString())\n
 is ULongArray -> result.append(element.contentToString())\n else ->
result.append(element.toString())\n }\n }\n result.append("]")\n
processed.removeAt(processed.lastIndex)\n}"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.collections\n\n/**\n * Data class representing a value from a collection or sequence, along with
its index in that collection or sequence.\n * @property value the underlying value.\n * @property index the
index of the value in the collection or sequence.\n */\n\npublic data class IndexedValue<out T>(public val index: Int,
public val value: T)\n"/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("MapAccessorsKt")\n\npackage
kotlin.collections\n\nimport kotlin.reflect.KProperty\nimport kotlin.internal.Exact\n\n/**\n * Returns the value of
the property for the given object from this read-only map.\n * @param thisRef the object for which the value is
requested
(not used).\n * @param property the metadata for the property, used to get the name of property and lookup the
value corresponding to this name in the map.\n * @return the property value.\n */\n * @throws
NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit
default (see [withDefault]).\n */\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> Map<in
String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n
@Suppress("UNCHECKED_CAST") (getOrNull(property.name) as V1)\n\n/**\n * Returns the value
of the property for the given object from this mutable map.\n * @param thisRef the object for which the value is
requested (not used).\n * @param property the metadata for the property, used to get the name of property and
lookup the value corresponding to this name in the map.\n * @return the property value.\n */\n * @throws

```

NoSuchElementException when the map doesn't contain value

for the property name and doesn't provide an implicit default (see [withDefault]).

```
\n@kotlin.jvm.JvmName("getVar")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V>\nMutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<>): V1 =\n@Suppress("UNCHECKED_CAST") (getOrNullImplicitDefault(property.name) as V1)\n\n/*\n * Stores the value of\n the property for the given object in this mutable map.\n * @param thisRef the object for which the value is\n requested (not used).\n * @param property the metadata for the property, used to get the name of property and store\n the value associated with that name in the map.\n * @param value the value to set.\n
```

```
\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef:\nAny?, property: KProperty<>, value: V) {\n this.put(property.name, value)\n}\n\n"/*\n * Copyright 2010-2018\n JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
```

source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```
\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage\nkotlin.collections\n\n/\n * Returns the value for the given key, or the implicit default value for this map.\n * By\n default no implicit value is provided for maps and a [NoSuchElementException] is thrown.\n * To create a map with\n implicit default value use [withDefault] method.\n * @throws NoSuchElementException when the map doesn't\n contain a value for the specified key and no implicit default was provided for that map.\n
```

```
\n@kotlin.jvm.JvmName("getOrNullImplicitDefaultNullable")\n@PublishedApi\ninternal fun <K, V> Map<K,\nV>.getOrNullImplicitDefault(key: K): V {\n if (this is MapWithDefault)\n return\nthis.getOrNullImplicitDefault(key)\n return getOrElseNullable(key, { throw NoSuchElementException("Key $key\nis missing in the map.") })\n}\n\n/\n * Returns a wrapper of this read-only\n map, having the implicit default value provided with the specified function [defaultValue].\n * This implicit\n default value is used when the original map doesn't contain a value for the key specified\n * and a value is obtained\n with [Map.getValue] function, for example when properties are delegated to the map.\n * When this map already\n has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.\n
```

```
\npublic fun <K, V> Map<K, V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =\n when (this) {\n is MapWithDefault -> this.map.withDefault(defaultValue)\n else -> MapWithDefaultImpl(this, defaultValue)\n }\n\n/\n * Returns a wrapper of this mutable map, having the implicit default value provided with the specified\n function [defaultValue].\n * This implicit default value is used when the original map doesn't contain a value for\n the key specified\n * and a value is obtained with [Map.getValue] function,
```

for example when properties are delegated to the map. When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.

```
*\n@kotlin.jvm.JvmName("withDefaultMutable")\npublic fun <K, V> MutableMap<K,\nV>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =\n when (this) {\n is\nMutableMapWithDefault -> this.map.withDefault(defaultValue)\n else -> MutableMapWithDefaultImpl(this,\ndefaultValue)\n }\n\nprivate interface MapWithDefault<K, out V> : Map<K, V> {\n public val map: Map<K,\nV>\n public fun getOrNullImplicitDefault(key: K): V\n}\n\nprivate interface MutableMapWithDefault<K, V> :\nMutableMap<K, V>, MapWithDefault<K, V> {\n public override val map: MutableMap<K, V>\n}\n\nprivate class MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) :\nMapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean = map.equals(other)\n
```

```
 override fun hashCode(): Int = map.hashCode()\n override fun toString(): String = map.toString()\n override\nval size: Int get() = map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n override fun\ncontainsKey(key: K): Boolean = map.containsKey(key)\n override fun containsValue(value: @UnsafeVariance\nV): Boolean = map.containsValue(value)\n override fun get(key: K): V? = map.get(key)\n override val keys:\nSet<K> get() = map.keys\n override val values: Collection<V> get() = map.values\n override val entries:\nSet<Map.Entry<K, V>> get() = map.entries\n\n override fun getOrNullImplicitDefault(key: K): V =\nmap.getOrElseNullable(key, { default(key) })\n}\n\nprivate class MutableMapWithDefaultImpl<K, V>(public\noverride val map: MutableMap<K, V>, private val default: (key: K) -> V) : MutableMapWithDefault<K, V> {\n
```

```

override fun equals(other: Any?): Boolean = map.equals(other)\n override fun hashCode(): Int =
map.hashCode()\n override fun toString():
String = map.toString()\n override val size: Int get() = map.size\n override fun isEmpty(): Boolean =
map.isEmpty()\n override fun containsKey(key: K): Boolean = map.containsKey(key)\n override fun
containsValue(value: @UnsafeVariance V): Boolean = map.containsValue(value)\n override fun get(key: K): V?
= map.get(key)\n override val keys: MutableSet<K> get() = map.keys\n override val values:
MutableCollection<V> get() = map.values\n override val entries: MutableSet<MutableMap.MutableEntry<K,
V>> get() = map.entries\n\n override fun put(key: K, value: V): V? = map.put(key, value)\n override fun
remove(key: K): V? = map.remove(key)\n override fun putAll(from: Map<out K, V>) = map.putAll(from)\n
override fun clear() = map.clear()\n\n override fun getOrDefault(key: K): V = map.getOrElseNullable(key,
{ default(key) })\n}\n\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("\CollectionsKt")\n\npackage
kotlin.collections\n\n**\n * Removes a single instance of the specified element from this\n * collection, if it is
present.\n * \n * Allows to overcome type-safety restriction of `remove` that requires to pass an element of type
`E`.\n * \n * @return `true` if the element has been successfully removed; `false` if it was not present in the
collection.\n * \n @kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes T>
MutableCollection<out T>.remove(element: T): Boolean =\n @Suppress("\UNCHECKED_CAST") (this as
MutableCollection<T>).remove(element)\n\n**\n * Removes all of this collection's elements that are also
contained in the specified collection.\n * \n * Allows to overcome type-safety restriction of `removeAll` that requires
to pass a collection of type `Collection<E>`.\n * \n
* @return `true` if any of the specified elements was removed from the collection, `false` if the collection was not
modified.\n * \n @kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes T>
MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean =\n
@Suppress("\UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)\n\n**\n * Retains
only the elements in this collection that are contained in the specified collection.\n * \n * Allows to overcome type-
safety restriction of `retainAll` that requires to pass a collection of type `Collection<E>`.\n * \n * @return `true` if
any element was removed from the collection, `false` if the collection was not modified.\n
*\n @kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out
T>.retainAll(elements: Collection<T>): Boolean =\n @Suppress("\UNCHECKED_CAST") (this as
MutableCollection<T>).retainAll(elements)\n\n**\n * Adds the specified [element]
to this mutable collection.\n * \n @kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in
T>.plusAssign(element: T) {\n this.add(element)\n}\n\n**\n * Adds all elements of the given [elements]
collection to this mutable collection.\n * \n @kotlin.internal.InlineOnly\n\npublic inline operator fun <T>
MutableCollection<in T>.plusAssign(elements: Iterable<T>) {\n this.addAll(elements)\n}\n\n**\n * Adds all
elements of the given [elements] array to this mutable collection.\n * \n @kotlin.internal.InlineOnly\n\npublic inline
operator fun <T> MutableCollection<in T>.plusAssign(elements: Array<T>) {\n
this.addAll(elements)\n}\n\n**\n * Adds all elements of the given [elements] sequence to this mutable collection.\n
*\n @kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(elements:
Sequence<T>) {\n this.addAll(elements)\n}\n\n**\n * Removes a single instance of the specified [element] from
this mutable collection.\n
*\n @kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(element:
T) {\n this.remove(element)\n}\n\n**\n * Removes all elements contained in the given [elements] collection from
this mutable collection.\n * \n @kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in
T>.minusAssign(elements: Iterable<T>) {\n this.removeAll(elements)\n}\n\n**\n * Removes all elements
contained in the given [elements] array from this mutable collection.\n * \n @kotlin.internal.InlineOnly\n\npublic
inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Array<T>) {\n

```

```

this.removeAll(elements)\n\n/**\n * Removes all elements contained in the given [elements] sequence from this
mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in
T>.minusAssign(elements: Sequence<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Adds all elements of the
given [elements] collection
to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements: Iterable<T>):
Boolean {\n when (elements) {\n is Collection -> return addAll(elements)\n else -> {\n var result:
Boolean = false\n for (item in elements)\n if (add(item)) result = true\n return result\n }\n }\n}\n\n/**\n * Adds all elements of the given [elements] sequence to this [MutableCollection].\n */\npublic fun
<T> MutableCollection<in T>.addAll(elements: Sequence<T>): Boolean {\n var result: Boolean = false\n for
(item in elements) {\n if (add(item)) result = true\n }\n return result\n}\n\n/**\n * Adds all elements of the
given [elements] array to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements:
Array<out T>): Boolean {\n return addAll(elements.asList())\n}\n\n/**\n * Converts this [Iterable] to a list if it is
not a [Collection].\n */\n * Otherwise, returns
this.\n */\ninternal fun <T> Iterable<T>.convertToListIfNotCollection(): Collection<T> =\n if (this is Collection)
this else toList()\n\n/**\n * Removes all elements from this [MutableCollection] that are also contained in the given
[elements] collection.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements: Iterable<T>): Boolean
{\n return removeAll(elements.convertToListIfNotCollection())\n}\n\n/**\n * Removes all elements from this
[MutableCollection] that are also contained in the given [elements] sequence.\n */\npublic fun <T>
MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {\n val list = elements.toList()\n return
list.isNotEmpty() && removeAll(list)\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are
also contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements:
Array<out T>): Boolean {\n return elements.isNotEmpty() && removeAll(elements.asList())\n}\n\n/**\n * Retains
only elements of this [MutableCollection] that are contained in the given [elements] collection.\n */\npublic fun
<T> MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {\n return
retainAll(elements.convertToListIfNotCollection())\n}\n\n/**\n * Retains only elements of this [MutableCollection]
that are contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in T>.retainAll(elements:
Array<out T>): Boolean {\n if (elements.isNotEmpty())\n return retainAll(elements.asList())\n else\n return retainNothing()\n}\n\n/**\n * Retains only elements of this [MutableCollection] that are contained in the
given [elements] sequence.\n */\npublic fun <T> MutableCollection<in T>.retainAll(elements: Sequence<T>):
Boolean {\n val list = elements.toList()\n if (list.isNotEmpty())\n return retainAll(list)\n else\n return
retainNothing()\n}\n\nprivate fun MutableCollection<*>.retainNothing(): Boolean {\n val result
= isEmpty()\n clear()\n return result\n}\n\n/**\n * Removes all elements from this [MutableIterable] that
match the given [predicate].\n */\n * @return `true` if any element was removed from this collection, or `false` when
no elements were removed and collection was not modified.\n */\npublic fun <T>
MutableIterable<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)\n\n/**\n * Retains only elements of this [MutableIterable] that match the given [predicate].\n */\n * @return `true` if any
element was removed from this collection, or `false` when all elements were retained and collection was not
modified.\n */\npublic fun <T> MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean =
filterInPlace(predicate, false)\n\nprivate fun <T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean,
predicateResultToRemove: Boolean): Boolean {\n var result = false\n with(iterator()) {\n while
(hasNext())\n if (predicate(next())
== predicateResultToRemove) {\n remove()\n result = true\n }\n }\n return
result\n}\n\n/**\n * Removes the element at the specified [index] from this list.\n */\n * In Kotlin one should use the
[MutableList.removeAt] function instead.\n */\n * @Deprecated("Use removeAt(index) instead.")\nReplaceWith("removeAt(index)"), level = DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\npublic inline
fun <T> MutableList<T>.remove(index: Int): T = removeAt(index)\n\n/**\n * Removes the first element from this
mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.\n */

```

```

*^@SinceKotlin("1.4")^@WasExperimental(ExperimentalStdlibApi::class)^public fun <T>
MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(0)^n/^n/**^n * Removes the first element from this mutable list and returns that removed element, or
returns `null` if this list is
empty.^n *^@SinceKotlin("1.4")^@WasExperimental(ExperimentalStdlibApi::class)^public fun <T>
MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)^n/^n/**^n * Removes the last
element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is
empty.^n *^@SinceKotlin("1.4")^@WasExperimental(ExperimentalStdlibApi::class)^public fun <T>
MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(lastIndex)^n/^n/**^n * Removes the last element from this mutable list and returns that removed element,
or returns `null` if this list is empty.^n
*^@SinceKotlin("1.4")^@WasExperimental(ExperimentalStdlibApi::class)^public fun <T>
MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)^n/^n/**^n * Removes all
elements from this [MutableList] that match the given [predicate].^n *^n * @return `true` if any element was
removed from this
collection, or `false` when no elements were removed and collection was not modified.^n *^npublic fun <T>
MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)^n/^n/**^n * Retains
only elements of this [MutableList] that match the given [predicate].^n *^n * @return `true` if any element was
removed from this collection, or `false` when all elements were retained and collection was not modified.^n
*^npublic fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate,
false)^n^nprivate fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove:
Boolean): Boolean {^n if (this !is RandomAccess)^n return (this as
MutableIterable<T>).filterInPlace(predicate, predicateResultToRemove)^n^n var writeIndex: Int = 0^n for
(readIndex in 0..lastIndex) {^n val element = this[readIndex]^n if (predicate(element) ==
predicateResultToRemove)^n continue^n
if (writeIndex != readIndex)^n this[writeIndex] = element^n writeIndex++^n }^n if (writeIndex
< size) {^n for (removeIndex in lastIndex downTo writeIndex)^n removeAt(removeIndex)^n return
true^n } else {^n return false^n }^n}^n",/*^n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming
Language contributors.^n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.^n *^n// Auto-generated file. DO NOT EDIT!^n^package kotlin.collections^n/^n/** An
iterator over a sequence of values of type `Byte`.^npublic abstract class ByteIterator : Iterator<Byte> {^n
override final fun next() = nextByte()^n^n /** Returns the next value in the sequence without boxing. *^n public
abstract fun nextByte(): Byte^n}^n/^n/** An iterator over a sequence of values of type `Char`. *^npublic abstract
class CharIterator : Iterator<Char> {^n override final
fun next() = nextChar()^n^n /** Returns the next value in the sequence without boxing. *^n public abstract fun
nextChar(): Char^n}^n/^n/** An iterator over a sequence of values of type `Short`. *^npublic abstract class
ShortIterator : Iterator<Short> {^n override final fun next() = nextShort()^n^n /** Returns the next value in the
sequence without boxing. *^n public abstract fun nextShort(): Short^n}^n/^n/** An iterator over a sequence of
values of type `Int`. *^npublic abstract class IntIterator : Iterator<Int> {^n override final fun next() = nextInt()^n^n
/** Returns the next value in the sequence without boxing. *^n public abstract fun nextInt(): Int^n}^n/^n/** An
iterator over a sequence of values of type `Long`. *^npublic abstract class LongIterator : Iterator<Long> {^n
override final fun next() = nextLong()^n^n /** Returns the next value in the sequence without boxing. *^n public
abstract fun nextLong(): Long^n}^n/^n/** An iterator over a sequence
of values of type `Float`. *^npublic abstract class FloatIterator : Iterator<Float> {^n override final fun next() =
nextFloat()^n^n /** Returns the next value in the sequence without boxing. *^n public abstract fun nextFloat():
Float^n}^n/^n/** An iterator over a sequence of values of type `Double`. *^npublic abstract class DoubleIterator :
Iterator<Double> {^n override final fun next() = nextDouble()^n^n /** Returns the next value in the sequence
without boxing. *^n public abstract fun nextDouble(): Double^n}^n/^n/** An iterator over a sequence of values of

```



```

type `Boolean`. */\npublic abstract class BooleanIterator : Iterator<Boolean> {\n override final fun next() =
nextBoolean()\n\n /** Returns the next value in the sequence without boxing. */\n public abstract fun
nextBoolean(): Boolean\n}\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nprivate open class ReversedListReadOnly<out T>(private val delegate: List<T>) :
AbstractList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n}\n\nprivate class ReversedList<T>(private val delegate: MutableList<T>) :
AbstractMutableList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n\n override fun clear() = delegate.clear()\n override fun removeAt(index:
Int): T = delegate.removeAt(reverseElementIndex(index))\n\n override fun set(index: Int, element: T): T =
delegate.set(reverseElementIndex(index), element)\n\n override fun add(index: Int, element: T) {\n
delegate.add(reversePositionIndex(index), element)\n }\n}\n\nprivate fun List<*>.reverseElementIndex(index:
Int) =\n if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element index
$index must be in range [${0..lastIndex}].")\n\nprivate fun List<*>.reversePositionIndex(index: Int) =\n if (index
in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range
[${0..size}].")\n}\n\n\n/*\n * Returns a reversed read-only view of the original List.\n * All changes made in the
original list will be reflected in the reversed one.\n * @sample samples.collections.ReversedViews.asReversedList\n
/\n\npublic fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)\n\n\n/\n * Returns a reversed
mutable view of the original mutable List.\n * All changes made in the original list will be reflected in the reversed
one and vice versa.\n * @sample samples.collections.ReversedViews.asReversedMutableList\n
*/\n\n@kotlin.jvm.JvmName("asReversedMutable")\n\npublic fun <T> MutableList<T>.asReversed():
MutableList<T> = ReversedList(this)\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\n@file:OptIn(Experimenta
lTypeInference::class)\n\npackage kotlin.sequences\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.experimental.ExperimentalTypeInference\n\n\n/*\n * Builds a
[Sequence] lazily yielding values one by one.\n * @see kotlin.sequences.generateSequence\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*/\n\n@SinceKotlin("1.3")\n\n@Suppress("DEPRECATION")\n\npublic fun <T> sequence(@BuilderInference block:
suspend SequenceScope<T>().->Unit): Sequence<T> = Sequence { iterator(block) }\n\n\n/*\n * Builds
an [Iterator] lazily yielding values one by one.\n * @sample
samples.collections.Sequences.Building.buildIterator\n * @sample samples.collections.Iterables.Building.iterable\n
*/\n\n@SinceKotlin("1.3")\n\n@Suppress("DEPRECATION")\n\npublic fun <T> iterator(@BuilderInference block:
suspend SequenceScope<T>().->Unit): Iterator<T> {\n val iterator = SequenceBuilderIterator<T>()\n
iterator.nextStep = block.createCoroutineUnintercepted(receiver = iterator, completion = iterator)\n return
iterator\n}\n\n\n/*\n * The scope for yielding values of a [Sequence] or an [Iterator], provides [yield] and [yieldAll]
suspension functions.\n * @see sequence\n * @see iterator\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*/\n\n@RestrictsSuspension\n\n@SinceKotlin("1.3")\n\npublic abstract class SequenceScope<in T> internal
constructor() {\n /**\n * Yields a value to the [Iterator]
being built and suspends\n * until the next value is requested.\n */\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n */\n public abstract suspend fun yield(value:

```

```

T)\n\n /**\n * Yields all values from the `iterator` to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * The sequence of values returned by the given iterator can be potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public abstract suspend fun yieldAll(iterator: Iterator<T>)\n\n /**\n * Yields a collections of values to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun yieldAll(elements: Iterable<T>) {\n if (elements is Collection && elements.isEmpty()) return\n return yieldAll(elements.iterator())\n }\n\n /**\n * Yields potentially infinite sequence of values to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * The sequence can be potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun yieldAll(sequence: Sequence<T>) = yieldAll(sequence.iterator())\n}\n\nprivate typealias State = Int\nprivate const val State_NotReady: State = 0\nprivate const val State_ManyNotReady: State = 1\nprivate const val State_ManyReady: State = 2\nprivate const val State_Ready: State = 3\nprivate const val State_Done: State = 4\nprivate const val State_Failed: State = 5\nprivate class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {\n private var state = State_NotReady\n private var nextValue: T? = null\n private var nextIterator: Iterator<T>? = null\n var nextStep: Continuation<Unit>? = null\n\n override fun hasNext(): Boolean {\n while (true) {\n when (state) {\n State_NotReady -> {}\n State_ManyNotReady -> {}\n if (nextIterator!!.hasNext()) {\n state = State_ManyReady\n return true\n }\n else {\n nextIterator = null\n State_Done -> return false\n State_Ready, State_ManyReady -> return true\n else -> throw exceptionalState()\n }\n state = State_Failed\n val step = nextStep!!\n nextStep = null\n step.resume(Unit)\n }\n }\n\n override fun next(): T {\n when (state) {\n State_NotReady, State_ManyNotReady -> return nextNotReady()\n State_ManyReady -> {\n state = State_ManyNotReady\n return nextIterator!!.next()\n }\n State_Ready -> {\n state = State_NotReady\n @Suppress("UNCHECKED_CAST")\n val result = nextValue as T\n nextValue = null\n return result\n }\n else -> throw exceptionalState()\n }\n }\n\n private fun nextNotReady(): T {\n if (!hasNext()) throw NoSuchElementException() else return next()\n }\n\n private fun exceptionalState(): Throwable = when (state) {\n State_Done -> NoSuchElementException()\n State_Failed -> IllegalStateException("Iterator has failed.")\n else -> IllegalStateException("Unexpected state of the iterator: $state")\n }\n\n override suspend fun yield(value: T) {\n nextValue = value\n state = State_Ready\n return suspendCoroutineUninterceptedOrReturn {\n c -> {\n nextStep = c\n COROUTINE_SUSPENDED\n }\n }\n }\n\n override suspend fun yieldAll(iterator: Iterator<T>) {\n if (!iterator.hasNext()) return\n nextIterator = iterator\n state = State_ManyReady\n return suspendCoroutineUninterceptedOrReturn {\n c -> {\n nextStep = c\n COROUTINE_SUSPENDED\n }\n }\n }\n\n // Completion continuation implementation\n override fun resumeWith(result: Result<Unit>) {\n result.getOrNull() // just rethrow exception if it is there\n state = State_Done\n }\n\n override val context: CoroutineContext\n get() = EmptyCoroutineContext\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun checkWindowSizeStep(size: Int, step: Int) {\n require(size > 0 && step > 0) {\n if (size != step)\n "\"Both size $size and step $step must be greater than zero.\"\n else\n \"$size $size must be greater than zero.\"\n }\n}\n\ninternal fun <T> Sequence<T>.windowedSequence(size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean): Sequence<List<T>> {\n checkWindowSizeStep(size, step)\n return Sequence { windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }\n}\n\ninternal fun <T> windowedIterator(iterator: Iterator<T>, size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean): Iterator<List<T>> {\n if (!iterator.hasNext()) return

```

```

EmptyIterator\n return iterator<List<T>> {\n val bufferInitialCapacity = size.coerceAtMost(1024)\n val gap = step - size\n if (gap >= 0) {\n var buffer = ArrayList<T>(bufferInitialCapacity)\n var skip = 0\n for (e in iterator) {\n if (skip > 0) { skip -= 1; continue }\n buffer.add(e)\n if (buffer.size == size) {\n yield(buffer)\n if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)\n skip = gap\n }\n }\n if (buffer.isNotEmpty()) {\n if (partialWindows || buffer.size == size)\n yield(buffer)\n } else {\n var buffer = RingBuffer<T>(bufferInitialCapacity)\n for (e in iterator) {\n buffer.add(e)\n if (buffer.isFull()) {\n if (buffer.size < size) {\n buffer =\n buffer.expanded(maxCapacity = size); continue\n }\n yield(if (reuseBuffer) buffer else\n ArrayList(buffer))\n buffer.removeFirst(step)\n }\n }\n if (partialWindows) {\n while (buffer.size > step) {\n yield(if (reuseBuffer)\n buffer else ArrayList(buffer))\n buffer.removeFirst(step)\n }\n if (buffer.isNotEmpty()) yield(buffer)\n }\n }\n }\n }\n\ninternal class MovingSubList<out E>(private val list: List<E>): AbstractList<E>(), RandomAccess {\n private var fromIndex: Int = 0\n private var _size: Int = 0\n\n fun move(fromIndex: Int, toIndex: Int) {\n checkRangeIndexes(fromIndex, toIndex, list.size)\n this.fromIndex = fromIndex\n this._size = toIndex - fromIndex\n }\n\n override fun get(index: Int): E {\n checkElementIndex(index, _size)\n return list[fromIndex + index]\n }\n\n override val size: Int get() = _size\n}\n\n/**\n * Provides ring buffer implementation.\n * Buffer overflow is not allowed so [add] doesn't\n * overwrite tail but raises an exception.\n */\nprivate class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int): AbstractList<T>(), RandomAccess {\n init {\n require(filledSize >= 0) { \"ring buffer filled size should not be negative but it is $filledSize\" }\n require(filledSize <= buffer.size) { \"ring buffer filled size: $filledSize cannot be larger than the buffer size: ${buffer.size}\" }\n }\n\n constructor(capacity: Int): this(arrayOfNulls<Any?>(capacity), 0)\n\n private val capacity = buffer.size\n private var startIndex: Int = 0\n override var size: Int = filledSize\n private set\n\n override fun get(index: Int): T {\n checkElementIndex(index, size)\n @Suppress(\"UNCHECKED_CAST\")\n return buffer[startIndex.forward(index)] as T\n }\n\n fun isFull() = size == capacity\n\n override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {\n private var count = size\n private var index = startIndex\n\n override fun computeNext() {\n if (count == 0) {\n done()\n } else {\n @Suppress(\"UNCHECKED_CAST\")\n setNext(buffer[index] as T)\n index = index.forward(1)\n count--\n }\n }\n }\n\n @Suppress(\"UNCHECKED_CAST\")\n override fun <T> toArray(array: Array<T>): Array<T> {\n val result: Array<T?> =\n if (array.size < this.size) array.copyOf(this.size) else array as Array<T?>\n val size = this.size\n var widx = 0\n var idx = startIndex\n while (widx < size && idx < capacity) {\n result[widx] = buffer[idx] as T\n widx++\n idx++\n }\n idx = 0\n while (widx < size) {\n result[widx] = buffer[idx] as T\n widx++\n idx++\n }\n if (result.size > this.size) result[this.size] = null\n return result as Array<T>\n }\n\n override fun toArray(): Array<Any?> {\n return toArray(arrayOfNulls(size))\n }\n\n /**\n * Creates a new ring buffer with the capacity equal to the minimum of [maxCapacity] and 1.5 * [capacity].\n * The returned ring buffer contains the same elements as this ring buffer.\n */\n fun expanded(maxCapacity: Int): RingBuffer<T> {\n val newCapacity = (capacity + (capacity shr 1) + 1).coerceAtMost(maxCapacity)\n val newBuffer = if (startIndex == 0) buffer.copyOf(newCapacity) else toArray(arrayOfNulls(newCapacity))\n return RingBuffer(newBuffer, size)\n }\n\n /**\n * Add [element] to the buffer or fail with [IllegalStateException] if no free space available in the buffer.\n */\n fun add(element: T) {\n if (isFull()) {\n throw IllegalStateException(\"ring buffer is full\")\n }\n buffer[startIndex.forward(size)] = element\n size++\n }\n\n /**\n * Removes [n] first elements from the buffer or fails with [IllegalArgumentException] if not enough elements in the buffer to remove.\n */\n fun removeFirst(n: Int) {\n require(n >= 0) { \"n shouldn't be negative but it is $n\" }\n require(n <= size) { \"n shouldn't be greater than the buffer size: n = $n, size = $size\" }\n if (n > 0) {\n val start = startIndex\n val

```



```

kotlin.internal.InlineOnly\n\n/**\n * Compares this object with the specified object for order. Returns zero if this
object is equal\n * to the specified [other] object, a negative number if it's less than [other], or a positive number\n *
if it's greater than [other].\n *\n * This function delegates to [Comparable.compareTo] and allows to call it in infix
form.\n */\n@InlineOnly\n@SinceKotlin("1.6")\npublic inline infix fun <T> Comparable<T>.compareTo(other:
T): Int =\n this.compareTo(other)\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.contracts\n\nimport
kotlin.internal.ContractsDsl\nimport kotlin.internal.InlineOnly\n\n/**\n * This marker distinguishes the
experimental contract declaration API and is used to opt-in for that feature\n * when declaring contracts of user
functions.\n *\n * Any usage of a declaration annotated with `@ExperimentalContracts` must be accepted either
by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalContracts::class)`,\n * or by
using the compiler argument `-opt-in=kotlin.contracts.ExperimentalContracts`.\n
*/\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.3")\n@RequiresOptIn\n@MustBeDocumente
d\npublic annotation class ExperimentalContracts\n\n/**\n * Provides a scope, where the functions of the contract
DSL, such as [returns], [callsInPlace], etc.,\n * can be used to
describe the contract of a function.\n *\n * This type is used as a receiver type of the lambda function passed to the
[contract] function.\n *\n * @see contract\n
*/\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin("1.3")\npublic interface ContractBuilder {\n /**\n
 * Describes a situation when a function returns normally, without any exceptions thrown.\n *\n * Use
[SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n *\n */\n //
@sample samples.contracts.returnsContract\n @ContractsDsl public fun returns(): Returns\n\n /**\n
 * Describes a situation when a function returns normally with the specified return [value].\n *\n * The possible
values of [value] are limited to `true`, `false` or `null`.\n *\n * Use [SimpleEffect.implies] function to describe a
conditional effect that happens in such case.\n *\n */\n // @sample samples.contracts.returnsTrueContract\n
// @sample samples.contracts.returnsFalseContract\n
// @sample samples.contracts.returnsNullContract\n @ContractsDsl public fun returns(value: Any?):
Returns\n\n /**\n
 * Describes a situation when a function returns normally with any value that is not `null`.\n *\n * Use
[SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n *\n */\n //
@sample samples.contracts.returnsNotNullContract\n @ContractsDsl public fun returnsNotNull():
ReturnsNotNull\n\n /**\n
 * Specifies that the function parameter [lambda] is invoked in place.\n *\n * This
contract specifies that:\n * 1. the function [lambda] can only be invoked during the call of the owner function,\n
 * and it won't be invoked after that owner function call is completed;\n * 2. _(optionally)_ the function [lambda]
is invoked the amount of times specified by the [kind] parameter,\n * see the [InvocationKind] enum for possible
values.\n *\n * A
function declaring the `callsInPlace` effect must be _inline_.\n *\n */\n /* @sample
samples.contracts.callsInPlaceAtMostOnceContract\n * @sample
samples.contracts.callsInPlaceAtLeastOnceContract\n * @sample
samples.contracts.callsInPlaceExactlyOnceContract\n * @sample
samples.contracts.callsInPlaceUnknownContract\n */\n @ContractsDsl public fun <R> callsInPlace(lambda:
Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace\n}\n\n/**\n * Specifies how
many times a function invokes its function parameter in place.\n *\n * See [ContractBuilder.callsInPlace] for the
details of the call-in-place function contract.\n
*/\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin("1.3")\npublic enum class InvocationKind {\n
/**\n * A function parameter will be invoked one time or not invoked at all.\n */\n // @sample
samples.contracts.callsInPlaceAtMostOnceContract\n @ContractsDsl AT_MOST_ONCE,\n\n /**\n * A
function parameter
will be invoked one or more times.\n */\n // @sample
samples.contracts.callsInPlaceAtLeastOnceContract\n @ContractsDsl AT_LEAST_ONCE,\n\n /**\n * A

```

function parameter will be invoked exactly one time.

```

 *
 *^
 // @sample
samples.contracts.callsInPlaceExactlyOnceContract
 @ContractsDsl EXACTLY_ONCE,
 /**
 * A
function parameter is called in place, but it's unknown how many times it can be called.
 *
 *^
 // @sample
samples.contracts.callsInPlaceUnknownContract
 @ContractsDsl UNKNOWN
}
}
/**
* Specifies the
contract of a function.
*
* The contract description must be at the beginning of a function and have at least one
effect.
*
* Only the top-level functions can have a contract for now.
*
* @param builder the lambda where
the contract of a function is described with the help of the [ContractBuilder] members.
*
*^
*/
@sample
samples.contracts.returnsContract
* @sample samples.contracts.returnsTrueContract
*
@sample samples.contracts.returnsFalseContract
* @sample samples.contracts.returnsNullContract
* @sample
samples.contracts.returnsNotNullContract
* @sample samples.contracts.callsInPlaceAtMostOnceContract
*
@sample samples.contracts.callsInPlaceAtLeastOnceContract
* @sample
samples.contracts.callsInPlaceExactlyOnceContract
* @sample
samples.contracts.callsInPlaceUnknownContract
*^
@ContractsDsl
@ExperimentalContracts
@InlineOnly
@
SinceKotlin("1.3")
@Suppress("UNUSED_PARAMETER")
public inline fun contract(builder:
ContractBuilder.() -> Unit) { }
}
"/
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
*^
package kotlin.coroutines
}
}
/**
* Marks coroutine context element that
intercepts coroutine continuations.
*
* The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the
interceptor
and
* intercepts all coroutine continuations with [interceptContinuation] invocations.
*
* [ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that
* its implementation delegates [get][CoroutineContext.Element.get] and
[minusKey][CoroutineContext.Element.minusKey]
* to [getPolymorphicElement] and [minusPolymorphicKey]
respectively.
*
* [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either
[ContinuationInterceptor.Key]
* or subtype key if it extends [AbstractCoroutineContextKey].
*^
@SinceKotlin("1.3")
public interface ContinuationInterceptor : CoroutineContext.Element {
*
* The key that defines *the* context interceptor.
*
*^
companion object Key :
CoroutineContext.Key<ContinuationInterceptor>
}
/**
* Returns continuation that wraps the original
[continuation], thus intercepting all resumptions.
*
* This function is invoked by coroutines framework
when needed and the resulting continuations are
* cached internally per each instance of the original
[continuation].
*
* This function may simply return original [continuation] if it does not want to intercept
this particular continuation.
*
* When the original [continuation] completes, coroutine framework invokes
[releaseInterceptedContinuation]
* with the resulting continuation if it was intercepted, that is if
`interceptContinuation` had previously
* returned a different continuation instance.
*^
public fun <T>
interceptContinuation(continuation: Continuation<T>): Continuation<T>
}
/**
* Invoked for the
continuation instance returned by [interceptContinuation] when the original
* continuation completes and will
not be used anymore. This function is invoked only if [interceptContinuation]
* had returned a different
continuation instance from the one it was invoked with.
*
* Default implementation
does nothing.
*
* @param continuation Continuation instance returned by this interceptor's
[interceptContinuation] invocation.
*^
public fun releaseInterceptedContinuation(continuation:
Continuation<*>) {
/* do nothing by default
*^
}
}
public override operator fun <E :
CoroutineContext.Element> get(key: CoroutineContext.Key<E>): E? {
/* getPolymorphicKey specialized for
ContinuationInterceptor key
* @OptIn(ExperimentalStdlibApi::class)
if (key is
AbstractCoroutineContextKey<*, *>) {
/* @Suppress("UNCHECKED_CAST")
return if
(key.isSubKey(this.key)) key.tryCast(this) as? E else null
}
/* @Suppress("UNCHECKED_CAST")
return if (ContinuationInterceptor === key) this as E else null
}
}
}
public override fun minusKey(key:
CoroutineContext.Key<*>): CoroutineContext {
/* minusPolymorphicKey specialized for
ContinuationInterceptor key
* @OptIn(ExperimentalStdlibApi::class)
}
}

```

```

 if (key is AbstractCoroutineContextKey<*, *>) {
 return if (key.isSubKey(this.key) &&
 key.tryCast(this) != null) EmptyCoroutineContext else this
 }
 return if (ContinuationInterceptor ===
 key) EmptyCoroutineContext else this
}

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
 * Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */
package kotlin.coroutines
/**
 * Persistent context for the
 * coroutine. It is an indexed set of [Element] instances.
 * An indexed set is a mix between a set and a map.
 * Every element in this set has a unique [Key].
 */
@SinceKotlin("1.3")
public interface CoroutineContext {
 /**
 * Returns the element with the given [key] from this context or `null`.
 */
 public operator fun <E> :
 Element? get(key: Key<E>): E?
 /**
 * Accumulates entries
 * of this context starting with [initial] value and applying [operation]
 * from left to right to current accumulator
 * value and each element of this context.
 */
 public fun <R> fold(initial: R, operation: (R, Element) -> R):
 R
 /**
 * Returns a context containing elements from this context and elements from
 * other [context].
 * The elements from this context with the same key as in the other one are
 * dropped.
 */
 public operator fun
 plus(context: CoroutineContext): CoroutineContext =
 if (context === EmptyCoroutineContext) this else // fast
 path -- avoid lambda creation
 context.fold(this) { acc, element ->
 val removed =
 acc.minusKey(element.key)
 if (removed === EmptyCoroutineContext) element else {
 //
 make sure interceptor is always last in the context (and thus is
 fast to get when present)
 val interceptor
 = removed[ContinuationInterceptor]
 if (interceptor == null) CombinedContext(removed, element) else {
 val left =
 removed.minusKey(ContinuationInterceptor)
 if (left === EmptyCoroutineContext)
 CombinedContext(element, interceptor) else
 CombinedContext(CombinedContext(left, element),
 interceptor)
 }
 }
 }
 /**
 * Returns a context containing elements from this
 * context, but without an element with
 * the specified [key].
 */
 public fun minusKey(key: Key<*>):
 CoroutineContext
 /**
 * Key for the elements of [CoroutineContext]. [E] is a type of element with
 * this key.
 */
 public interface Key<E> : Element
 /**
 * An element of the [CoroutineContext]. An
 * element of the coroutine context is a singleton context by itself.
 */
 public interface Element :
 CoroutineContext
 /**
 * A key of this coroutine context element.
 */
 public val key: Key<*>
 public override operator fun <E> : Element? get(key: Key<E>): E?
 =
 @Suppress("UNCHECKED_CAST")
 if (this.key == key) this as E else null
 public
 override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
 operation(initial, this)
 public
 override fun minusKey(key: Key<*>): CoroutineContext =
 if (this.key == key) EmptyCoroutineContext
 else this
}

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */
import kotlin.coroutines.CoroutineContext.Element
import
 kotlin.coroutines.CoroutineContext.Key
/**
 * Base class for [CoroutineContext.Element] implementations.
 */
@SinceKotlin("1.3")
public abstract class AbstractCoroutineContextElement(
 public
 override val key: Key<*> : Element
) /**
 * Base class for [CoroutineContext.Key] associated with
 * polymorphic [CoroutineContext.Element] implementation.
 * Polymorphic element implementation implies
 * delegating its [get][Element.get] and [minusKey][Element.minusKey]
 * to [getPolymorphicElement] and
 * [minusPolymorphicKey] respectively.
 * Polymorphic elements can be extracted from the coroutine context
 * using both element key and its supertype key.
 * Example of polymorphic elements:
 */
` ` `
 * open class
 BaseElement : CoroutineContext.Element {
 * companion object Key : CoroutineContext.Key<BaseElement>
 * override val key: CoroutineContext.Key<*> get() = Key
 * // It is important to use getPolymorphicKey and
 * minusPolymorphicKey
 * override fun <E> : CoroutineContext.Element? get(key:
 CoroutineContext.Key<E>):
 E? = getPolymorphicElement(key)
 * override fun minusKey(key: CoroutineContext.Key<*>):
 CoroutineContext =
 minusPolymorphicKey(key)
 }
 *
 * class DerivedElement : BaseElement() {
 * companion object Key :
 AbstractCoroutineContextKey<BaseElement, DerivedElement>(
 BaseElement, { it as? DerivedElement })
 }
}

```





```

CombinedContext) {\n cur = next\n } else {\n return contains(next as Element)\n }\n }\n }\n\n override fun equals(other: Any?): Boolean =\n this === other || other is CombinedContext\n && other.size() == size() && other.containsAll(this)\n\n override fun hashCode(): Int = left.hashCode() +\n element.hashCode()\n\n override fun toString(): String =\n "[" + fold("") { acc, element ->\n if\n (acc.isEmpty()) element.toString() else "$acc, $element"\n } + "]" \n\n private fun writeReplace(): Any {\n val n = size()\n val elements = arrayOfNulls<CoroutineContext>(n)\n var index = 0\n fold(Unit) { _,\n element -> elements[index++]\n = element }\n check(index == n)\n @Suppress("UNCHECKED_CAST")\n return\n Serialized(elements as Array<CoroutineContext>)\n }\n\n private class Serialized(val elements:\n Array<CoroutineContext>) : Serializable {\n companion object {\n private const val serialVersionUID:\n Long = 0L\n }\n private fun readResolve(): Any = elements.fold(EmptyCoroutineContext,\n CoroutineContext::plus)\n }\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming\n Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the\n license/LICENSE.txt file.\n\n*/\n@file:kotlin.jvm.JvmName("IntrinsicsKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage\n kotlin.coroutines.intrinsics\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.*\nimport\n kotlin.internal.InlineOnly\n\n/**\n * Obtains the current continuation instance inside suspend functions and either\n suspends\n * currently running\n coroutine or returns result immediately without suspension.\n * If the [block] returns the special\n [COROUTINE_SUSPENDED] value, it means that suspend function did suspend the execution and will\n * not\n return any result immediately. In this case, the [Continuation] provided to the [block] shall be\n * resumed by\n invoking [Continuation.resumeWith] at some moment in the\n * future when the result becomes available to resume\n the computation.\n * Otherwise, the return value of the [block] must have a type assignable to [T] and represents\n the result of this suspend function.\n * It means that the execution was not suspended and the [Continuation]\n provided to the [block] shall not be invoked.\n * As the result type of the [block] is declared as `Any?` and cannot be\n correctly type-checked,\n * its proper return type remains on the conscience of the suspend function's author.\n * Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's thread without\n going through the\n * [ContinuationInterceptor] that might be present in the coroutine's [CoroutineContext].\n * It\n is the invoker's responsibility to ensure that a proper invocation context is established.\n * [Continuation.intercepted]\n can be used to acquire the intercepted continuation.\n\n * Note that it is not recommended to call either\n [Continuation.resume] nor [Continuation.resumeWithException] functions synchronously\n * in the same\n stackframe where suspension function is run. Use [suspendCoroutine] as a safer way to obtain current\n * continuation instance.\n\n*/\n@SinceKotlin("1.3")\n@InlineOnly\n@Suppress("UNUSED_PARAMETER",\n "RedundantSuspendModifier")\npublic suspend inline fun <T>\n suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n throw NotImplementedError("Implementation of\n suspendCoroutineUninterceptedOrReturn is intrinsic")\n }\n\n/**\n * This value is\n used as a return value of [suspendCoroutineUninterceptedOrReturn] `block` argument to state that\n * the execution\n was suspended and will not return any result immediately.\n * Note: this value should not be used in general\n code. Using it outside of the context of\n * `suspendCoroutineUninterceptedOrReturn` function return value\n (including, but not limited to,\n * storing this value in other properties, returning it from other functions, etc)\n * can\n lead to unspecified behavior of the code.\n\n*/\n// It is implemented as property with getter to avoid ProGuard\n <clinit> problem with multifile IntrinsicsKt class\n\n@SinceKotlin("1.3")\npublic val COROUTINE_SUSPENDED:\n Any get() = CoroutineSingletons.COROUTINE_SUSPENDED\n\n// Using enum here ensures two important\n properties:\n // 1. It makes SafeContinuation serializable with all kinds of serialization frameworks (since all of them\n natively support enums)\n // 2. It improves debugging experience, since you clearly see toString() value\n of those objects and what package they come from\n\n@SinceKotlin("1.3")\n@PublishedApi // This class is\n Published API via serialized representation of SafeContinuation, don't rename/move\n\ninternal enum class

```



contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.experimental\n\n/\*\*\n \* This annotation marks the experimental [ObjCName][kotlin.native.ObjCName] annotation.\n \*/\n\n@RequiresOptIn\n@Target(AnnotationTarget.ANNOTATION\_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@MustBeDocumented\npublic annotation class ExperimentalObjCName\n", "/\*\n \* Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.experimental\n\n/\*\*\n \* This annotation marks the experimental Objective-C export refinement annotations.\n \*/\n\n@RequiresOptIn\n@Target(AnnotationTarget.ANNOTATION\_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@MustBeDocumented\npublic annotation class ExperimentalObjCRefinement\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.experimental\n\n/\*\*\n \* Performs a bitwise AND operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()\n\n/\*\*\n \* Performs a bitwise OR operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()\n\n/\*\*\n \* Performs a bitwise XOR operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()\n\n/\*\*\n \* Inverts the bits in this value.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.inv(): Byte = (this.toInt().inv()).toByte()\n\n/\*\*\n \* Performs a bitwise AND operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.and(other: Short): Short = (this.toInt() and other.toInt()).toShort()\n\n/\*\*\n \* Performs a bitwise OR operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short = (this.toInt() or other.toInt()).toShort()\n\n/\*\*\n \* Performs a bitwise XOR operation between the two values.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short = (this.toInt() xor other.toInt()).toShort()\n\n/\*\*\n \* Inverts the bits in this value.\n \*/\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short = (this.toInt().inv()).toShort()\n\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.experimental\n\n/\*\*\n \* The experimental marker for type inference augmenting annotations.\n \* Any usage of a declaration annotated with `@ExperimentalTypeInference` must be accepted either by\n \* annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalTypeInference::class)`,\n \* or by using the compiler argument `-opt-in=kotlin.experimental.ExperimentalTypeInference`.\n \*/\n\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.ANNOTATION\_CLASS)\n@SinceKotlin("1.3")\npublic annotation class ExperimentalTypeInference\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.internal\n\n/\*\*\n \* Specifies that the corresponding type should be ignored during type inference.\n \*/\n\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class NoInfer\n\n/\*\*\n \* Specifies that the constraint built for the type during type inference should be an equality one.\n \*/\n\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class Exact\n\n/\*\*\n \* Specifies that a corresponding member has the lowest priority in overload resolution.\n \*/\n\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class

LowPriorityInOverloadResolution\n\n/\*\*\n

\* Specifies that the corresponding member has the highest priority in overload resolution. Effectively this means that\n \* an extension annotated with this annotation will win in overload resolution over a member with the same signature.\n \*/\n@Target(AnnotationTarget.FUNCTION,

AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class

HidesMembers\n\n/\*\*\n \* The value of this type parameter should be mentioned in input types (argument types, receiver type or expected type).\n

\*/\n@Target(AnnotationTarget.TYPE\_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class OnlyInputTypes\n\n/\*\*\n \* Specifies that this function should not be called directly without inlining\n \*/\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,

AnnotationTarget.PROPERTY\_GETTER,

AnnotationTarget.PROPERTY\_SETTER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class

InlineOnly\n\n/\*\*\n \* Specifies that this declaration can have dynamic receiver type.\n \*/\n@Target(AnnotationTarget.FUNCTION,

AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class

DynamicExtension\n\n/\*\*\n \* The value of this parameter should be a property reference expression (`this::foo`), referencing a `lateinit` property,\n \* the backing field of which is accessible at the point where the corresponding argument is passed.\n

\*/\n@Target(AnnotationTarget.VALUE\_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.2")\ninternal annotation class AccessibleLateinitPropertyLiteral\n\n/\*\*\n \* Specifies that this declaration is only completely supported since the specified version.\n \* The Kotlin compiler of an earlier version is going to report a diagnostic on usages of this declaration.\n \* The diagnostic message can be specified with [message], or via [errorCode] (takes less space, but might not be immediately clear\n \* to the user). The diagnostic severity can be specified with [level]: WARNING/ERROR

mean that either a warning or an error\n \* is going to be reported, HIDDEN means that the declaration is going to be removed from resolution completely.\n \* [versionKind] specifies which version should be compared with the [version] value, when compiling the usage of the annotated declaration.\n \* Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE\_VERSION] was supported, so the Kotlin compiler before 1.2 is going to\n \* treat any [RequireKotlin] as if it requires the language version. Since 1.2, the Kotlin compiler supports\n \* [RequireKotlinVersionKind.LANGUAGE\_VERSION], [RequireKotlinVersionKind.COMPILER\_VERSION] and [RequireKotlinVersionKind.API\_VERSION].\n \* If the actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),\n \* Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a requirement\n \* on the language version.\n \* This

annotation is erased at compile time; its arguments are stored in a more compact form in the Kotlin metadata.\n

\*/\n@Target(AnnotationTarget.CLASS, AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,

AnnotationTarget.CONSTRUCTOR,

AnnotationTarget.TYPEALIAS)\n@Retention(AnnotationRetention.SOURCE)\n@Repeatable\n@SinceKotlin("1.

2")\ninternal annotation class RequireKotlin(\n val version: String,\n val message: String = "",\n val level: DeprecationLevel = DeprecationLevel.ERROR,\n val versionKind: RequireKotlinVersionKind =

RequireKotlinVersionKind.LANGUAGE\_VERSION,\n val errorCode: Int = -1)\n\n/\*\*\n \* The kind of the version that is required by [RequireKotlin].\n \*/\n@SinceKotlin("1.2")\ninternal enum class

RequireKotlinVersionKind {\n LANGUAGE\_VERSION,\n COMPILER\_VERSION,\n

API\_VERSION,\n}\n\n/\*\*\n \* Specifies that this declaration is a part of special DSL, used for constructing function's contract.\n \*/\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.2")\ninternal

annotation class ContractsDsl\n\n/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\n\npackage kotlin.internal\n\n// a mod b (in arithmetical sense)\nprivate fun mod(a:

```

Int, b: Int): Int {\n val mod = a % b\n return if (mod >= 0) mod else mod + b\n}\n\nprivate fun mod(a: Long, b:
Long): Long {\n val mod = a % b\n return if (mod >= 0) mod else mod + b\n}\n\n// (a - b) mod c\nprivate fun
differenceModulo(a: Int, b: Int, c: Int): Int {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\nprivate fun
differenceModulo(a: Long, b: Long, c: Long): Long {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\n/**\n *
Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in
the range\n * from [start] to [end] in case of a positive [step], or from [end]
to [start] in case of a negative\n * [step].\n * No validation on passed parameters is performed. The given
parameters should satisfy the condition:\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >=
end`.\n * @param start first element of the progression\n * @param end ending bound for the progression\n *
@param step increment, or difference of successive elements in the progression\n * @return the final element of the
progression\n * @suppress\n */\n\n@PublishedApi\ninternal fun getProgressionLastElement(start: Int, end: Int, step:
Int): Int = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n step < 0 -> if
(start <= end) end else end + differenceModulo(start, end, -step)\n else -> throw
kotlin.IllegalArgumentException("Step is zero.")\n}\n\n/**\n * Calculates the final element of a bounded
arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start]
to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n * No validation on
passed parameters is performed. The given parameters should satisfy the condition:\n * - either `step > 0` and
`start <= end`,\n * - or `step < 0` and `start >= end`.\n * @param start first element of the progression\n *
@param end ending bound for the progression\n * @param step increment, or difference of successive elements in
the progression\n * @return the final element of the progression\n * @suppress\n */\n\n@PublishedApi\ninternal fun
getProgressionLastElement(start: Long, end: Long, step: Long): Long = when {\n step > 0 -> if (start >= end) end
else end - differenceModulo(end, start, step)\n step < 0 -> if (start <= end) end else end + differenceModulo(start,
end, -step)\n else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n\n"/**\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.properties\nimport kotlin.reflect.KProperty\n\n/**\n * Standard property delegates.\n
*/\n\npublic object Delegates {\n /**\n * Returns a property delegate for a read/write property with a non-`null`
value that is initialized not during\n * object construction time but at a later time. Trying to read the property
before the initial value has been\n * assigned results in an exception.\n *\n * @sample
samples.properties.Delegates.notNullDelegate\n */\n public fun <T : Any> notNull():
ReadWriteProperty<Any?, T> = NotNullVar()\n\n /**\n * Returns a property delegate for a read/write property
that calls a specified callback function when changed.\n * @param initialValue the initial value of the property.\n
 * @param onChange the callback which is called after the change of the property is made. The value of the
property\n *\n * has already been changed when this callback is invoked.\n *\n * @sample
samples.properties.Delegates.observableDelegate\n */\n public inline fun <T> observable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Unit):\n
ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n override fun
afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)\n
 }\n\n /**\n * Returns a property delegate for a read/write property that calls a specified callback function when
changed,\n * allowing the callback to veto the modification.\n * @param initialValue the initial value of the
property.\n * @param onChange the callback which is called before a change to the property value is attempted.\n
 * The value of the property hasn't been changed yet, when this callback is invoked.\n * If the callback returns
`true` the value of the property is being set to the new value,\n * and if the callback returns `false` the new value
is discarded and the property remains its old value.\n *\n * @sample
samples.properties.Delegates.vetoableDelegate\n */\n @sample
samples.properties.Delegates.throwVetoableDelegate\n */\n public inline fun <T> vetoable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):\n

```

```

ReadWriteProperty<Any?, T> = {
 object : ObservableProperty<T>(initialValue) {
 override fun
 beforeChange(property: KProperty<*>, oldValue: T, newValue: T): Boolean = onChange(property, oldValue,
 newValue)
 }
}

private class NotNullVar<T : Any>(): ReadWriteProperty<Any?, T> {
 private var
 value: T? = null
 public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
 return value
 ?: throw IllegalStateException("Property ${property.name} should be initialized
 before get.")
 }
 public override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {
 this.value = value
 }
}

/* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
*/
package kotlin.properties
import kotlin.reflect.KProperty

/**
 * Base
 * interface that can be used for implementing property delegates of read-only properties.
 * This is provided only
 * for convenience; you don't have to extend this interface
 * as long as your property delegate has methods with the
 * same signatures.
 * @param T the type of object which owns the delegated property.
 * @param V the type of
 * the property value.
 */
public interface ReadOnlyProperty<in T, out V> {
 /**
 * Returns the value of
 * the property for the given object.
 * @param thisRef the object for which the value
 * is requested.
 * @param property the metadata for the property.
 * @return the property value.
 */
 public operator fun getValue(thisRef: T, property: KProperty<*>): V
}

/**
 * Base interface that can be used
 * for implementing property delegates of read-write properties.
 * This is provided only for convenience; you
 * don't have to extend this interface
 * as long as your property delegate has methods with the same signatures.
 * @param T the type of object which owns the delegated property.
 * @param V the type of the property value.
 */
public interface ReadWriteProperty<in T, V> : ReadOnlyProperty<T, V> {
 /**
 * Returns the value of
 * the property for the given object.
 * @param thisRef the object for which the value is requested.
 * @param
 * property the metadata for the property.
 * @return the property value.
 */
 public override operator fun
 getValue(thisRef: T, property: KProperty<*>): V
 /**
 * Sets
 * the value of the property for the given object.
 * @param thisRef the object for which the value is requested.
 * @param
 * property the metadata for the property.
 * @param value the value to set.
 */
 public operator
 fun setValue(thisRef: T, property: KProperty<*>, value: V)
}

/**
 * Base interface that can be used for
 * implementing property delegate providers.
 * This is provided only for convenience; you don't have to extend
 * this interface
 * as long as your delegate provider has a method with the same signature.
 * @param T the type
 * of object which owns the delegated property.
 * @param D the type of property delegates this provider provides.
 */
@SinceKotlin("1.4")
public interface PropertyDelegateProvider<in T, out D> {
 /**
 * Returns the
 * delegate of the property for the given object.
 * This function can be used to extend the logic of creating
 * the object (e.g. perform validation checks)
 * to which the
 * property implementation is delegated.
 * @param thisRef the object for which property delegate is
 * requested.
 * @param property the metadata for the property.
 * @return the property delegate.
 */
 public operator fun provideDelegate(thisRef: T, property: KProperty<*>): D
}

/* Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.properties
import kotlin.reflect.KProperty

/**
 * Implements the core logic of a property delegate for a read/write property that
 * calls callback functions when changed.
 * @param initialValue the initial value of the property.
 */
public abstract class ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {
 private var value =
 initialValue
 /**
 * The callback which is called before a change to the property value
 * is attempted.
 * The value of the property hasn't been changed yet, when this callback is invoked.
 * If the
 * callback returns `true` the value of the property is being set to the new value,
 * and if the callback returns
 * `false` the new value is discarded and the property remains its old value.
 */
 protected open fun
 beforeChange(property: KProperty<*>, oldValue: V, newValue: V): Boolean = true
 /**
 * The callback
 * which is called after the change of the property is made. The value of the property
 * has already been changed
 * when this callback is invoked.
 */
 protected open fun afterChange(property: KProperty<*>, oldValue: V,
 newValue: V): Unit {}
 public override fun getValue(thisRef: Any?, property: KProperty<*>): V {

```

```

return value\n }\n public override fun setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n val
oldValue = this.value\n if (!beforeChange(property, oldValue, value)) {\n
 return\n }\n this.value = value\n afterChange(property, oldValue, value)\n }\n}"/**\n *
Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
/\n\n@file:Suppress("PackageDirectoryMismatch")\npackage kotlin\n\nimport kotlin.reflect.\n\n/**\n * An
extension operator that allows delegating a read-only property of type [V]\n * to a property reference to a property
of type [V] or its subtype.\n * @receiver A property reference to a read-only or mutable property of type [V] or
its subtype.\n * The reference is without a receiver, i.e. it either references a top-level property or\n * has the
receiver bound to it.\n * Example:\n * ```\n * class Login(val username: String)\n * val defaultLogin =
Login("Admin")\n * val defaultUsername by defaultLogin::username\n * // equivalent to\n * val defaultUserName
get() = defaultLogin.username\n * ```\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline
operator fun <V> KProperty0<V>.getValue(thisRef: Any?, property: KProperty<*>): V {\n return
get()\n}\n\n/**\n * An extension operator that allows delegating a mutable property of type [V]\n * to a property
reference to a mutable property of the same type [V].\n * @receiver A property reference to a mutable property
of type [V].\n * The reference is without a receiver, i.e. it either references a top-level property or\n * has the
receiver bound to it.\n * Example:\n * ```\n * class Login(val username: String, var
incorrectAttemptCounter: Int = 0)\n * val defaultLogin = Login("Admin")\n * var defaultLoginAttempts by
defaultLogin::incorrectAttemptCounter\n * // equivalent to\n * var defaultLoginAttempts: Int\n * get() =
defaultLogin.incorrectAttemptCounter\n * set(value) { defaultLogin.incorrectAttemptCounter = value }\n * ```\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline operator fun <V> KMutableProperty0<V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n
 set(value)\n}\n\n/**\n * An extension operator that allows delegating a read-only member or extension property of
type [V]\n * to a property reference to a member or extension property of type [V] or its subtype.\n * @receiver
A property reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound
receiver of type [T].\n * Example:\n * ```\n * class Login(val username: String)\n * val Login.user by
Login::username\n * // equivalent to\n * val Login.user get() = this.username\n * ```\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KProperty1<T,
V>.getValue(thisRef: T, property: KProperty<*>): V {\n return get(thisRef)\n}\n\n/**\n * An extension operator
that allows delegating a mutable member or extension property of type [V]\n *
to a property reference to a member or extension mutable property of the same type [V].\n * @receiver A
property reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound
receiver of type [T].\n * Example:\n * ```\n * class Login(val username: String, var
incorrectAttemptCounter: Int)\n * var Login.attempts by Login::incorrectAttemptCounter\n * // equivalent to\n * var
Login.attempts: Int\n * get() = this.incorrectAttemptCounter\n * set(value) { this.incorrectAttemptCounter =
value }\n * ```\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V>
KMutableProperty1<T, V>.setValue(thisRef: T, property: KProperty<*>, value: V) {\n set(thisRef,
value)\n}"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.random\n\nimport
kotlin.math.nextDown\n\n/**\n * An abstract class that is implemented by random number generator algorithms.\n
*\n * The companion object [Random.Default] is the default instance of [Random].\n * To get a seeded instance
of random generator use [Random] function.\n * @sample samples.random.Randoms.defaultRandom\n
*/\n@SinceKotlin("1.3")\npublic abstract class Random {\n /**\n * Gets the next random [bitCount] number
of bits.\n * Generates an `Int` whose lower [bitCount] bits are filled with random values and the remaining
upper bits are zero.\n * @param bitCount number of bits to generate, must be in range 0..32, otherwise the
behavior is unspecified.\n * @sample samples.random.Randoms.nextBits\n */\n public abstract fun
nextBits(bitCount: Int): Int\n /**\n * Gets the next random `Int` from the random number generator.\n */
}

```

```

* Generates an `Int` random value uniformly distributed between `Int.MIN_VALUE`
and `Int.MAX_VALUE` (inclusive).\n * \n * @sample samples.random.Randoms.nextInt\n * \n public
open fun nextInt(): Int = nextBits(32)\n\n /** \n * Gets the next random non-negative `Int` from the random
number generator less than the specified [until] bound.\n * \n * Generates an `Int` random value uniformly
distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n * \n * @param until must be
positive.\n * \n * @throws IllegalArgumentException if [until] is negative or zero.\n * \n * @sample
samples.random.Randoms.nextIntFromUntil\n * \n public open fun nextInt(until: Int): Int = nextInt(0, until)\n\n
/** \n * Gets the next random `Int` from the random number generator in the specified range.\n * \n *
Generates an `Int` random value uniformly distributed between the specified [from] (inclusive) and [until]
(exclusive) bounds.\n * \n * @throws IllegalArgumentException if [from]
is greater than or equal to [until].\n * \n * @sample samples.random.Randoms.nextIntFromUntil\n * \n
public open fun nextInt(from: Int, until: Int): Int {\n checkRangeBounds(from, until)\n val n = until -
from\n if (n > 0 || n == Int.MIN_VALUE) {\n val rnd = if (n and -n == n) {\n val bitCount =
fastLog2(n)\n nextBits(bitCount)\n } else {\n var v: Int\n do {\n val
bits = nextInt().ushr(1)\n v = bits % n\n } while (bits - v + (n - 1) < 0)\n v\n }\n return from + rnd\n } else {\n while (true) {\n val rnd = nextInt()\n if (rnd in
from until until) return rnd\n }\n }\n}\n\n /** \n * Gets the next random `Long` from the random
number generator.\n * \n * Generates a `Long` random value uniformly distributed
between `Long.MIN_VALUE` and `Long.MAX_VALUE` (inclusive).\n * \n * @sample
samples.random.Randoms.nextLong\n * \n public open fun nextLong(): Long = nextInt().toLong().shl(32) +
nextInt()\n\n /** \n * Gets the next random non-negative `Long` from the random number generator less than the
specified [until] bound.\n * \n * Generates a `Long` random value uniformly distributed between `0` (inclusive)
and the specified [until] bound (exclusive).\n * \n * @param until must be positive.\n * \n * @throws
IllegalArgumentException if [until] is negative or zero.\n * \n * @sample
samples.random.Randoms.nextLongFromUntil\n * \n public open fun nextLong(until: Long): Long =
nextLong(0, until)\n\n /** \n * Gets the next random `Long` from the random number generator in the specified
range.\n * \n * Generates a `Long` random value uniformly distributed between the specified [from] (inclusive)
and [until] (exclusive) bounds.\n
\n * \n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n * \n * @sample
samples.random.Randoms.nextLongFromUntil\n * \n public open fun nextLong(from: Long, until: Long): Long
{\n checkRangeBounds(from, until)\n val n = until - from\n if (n > 0) {\n val rnd: Long\n
if (n and -n == n) {\n val nLow = n.toInt()\n val nHigh = (n ushr 32).toInt()\n rnd =
when {\n nLow != 0 -> {\n val bitCount = fastLog2(nLow)\n //
toUInt().toLong()\n nextBits(bitCount).toLong() and 0xFFFF_FFFF\n }\n
nHigh == 1 ->{\n // toUInt().toLong()\n nextInt().toLong() and 0xFFFF_FFFF\n
 }\n else -> {\n val bitCount = fastLog2(nHigh)\n nextBits(bitCount).toLong().shl(32)
+ (nextInt().toLong() and 0xFFFF_FFFF)\n }\n } else {\n var v: Long\n
do {\n val bits = nextLong().ushr(1)\n v = bits % n\n } while (bits - v + (n - 1)
< 0)\n rnd = v\n }\n return from + rnd\n } else {\n while (true) {\n val
rnd = nextLong()\n if (rnd in from until until) return rnd\n }\n }\n }\n}\n\n /** \n * Gets the
next random [Boolean] value.\n * \n * @sample samples.random.Randoms.nextBoolean\n * \n public open
fun nextBoolean(): Boolean = nextBits(1) != 0\n\n /** \n * Gets the next random [Double] value uniformly
distributed between 0 (inclusive) and 1 (exclusive).\n * \n * @sample samples.random.Randoms.nextDouble\n
\n * \n public open fun nextDouble(): Double = doubleFromParts(nextBits(26), nextBits(27))\n\n
/** \n * Gets the next random non-negative `Double` from the random number generator less than the specified
[until] bound.\n * \n * Generates a `Double` random value uniformly distributed between 0 (inclusive) and
[until] (exclusive).\n * \n * @throws IllegalArgumentException if [until] is negative or zero.\n * \n *
@sample samples.random.Randoms.nextDoubleFromUntil\n * \n public open fun nextDouble(until: Double):

```



```

Double = nextDouble(0.0, until)\n\n /**\n * Gets the next random `Double` from the random number generator
in the specified range.\n *\n * Generates a `Double` random value uniformly distributed between the specified
[from] (inclusive) and [until] (exclusive) bounds.\n *\n * [from] and [until] must be finite otherwise the
behavior is unspecified.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to
[until].\n *\n * @sample samples.random.Randoms.nextDoubleFromUntil\n
*/\n public open fun nextDouble(from: Double, until: Double): Double {\n checkRangeBounds(from,
until)\n val size = until - from\n val r = if (size.isInfinite() && from.isFinite() && until.isFinite()) {\n
val r1 = nextDouble() * (until / 2 - from / 2)\n from + r1 + r1\n } else {\n from + nextDouble() *
size\n }\n return if (r >= until) until.nextDown() else r\n }\n\n /**\n * Gets the next random [Float]
value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n *\n * @sample
samples.random.Randoms.nextFloat\n
*/\n public open fun nextFloat(): Float = nextBits(24) / (1 shl
24).toFloat()\n\n /**\n * Fills a subrange of the specified byte [array] starting from [fromIndex] inclusive and
ending [toIndex] exclusive\n * with random bytes.\n *\n * @return [array] with the subrange filled with
random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n
*/\n public open fun nextBytes(array: ByteArray, fromIndex: Int = 0, toIndex: Int = array.size): ByteArray {\n
require(fromIndex in 0..array.size && toIndex in 0..array.size) { "\"fromIndex ($fromIndex) or toIndex ($toIndex)
are out of range: 0..${array.size}." }\n require(fromIndex <= toIndex) { "\"fromIndex ($fromIndex) must be not
greater than toIndex ($toIndex).\" }\n val steps = (toIndex - fromIndex) / 4\n var position = fromIndex\n
repeat(steps) {\n val v = nextInt()\n array[position] = v.toByte()\n array[position + 1] =
v.ushr(8).toByte()\n array[position + 2] = v.ushr(16).toByte()\n array[position + 3] =
v.ushr(24).toByte()\n position += 4\n }\n val remainder = toIndex - position\n val vr =
nextBits(remainder * 8)\n for (i in 0 until remainder) {\n array[position + i] = vr.ushr(i * 8).toByte()\n
}\n return
array\n }\n\n /**\n * Fills the specified byte [array] with random bytes and returns it.\n *\n * @return
[array] filled with random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n
*/\n public open
fun nextBytes(array: ByteArray): ByteArray = nextBytes(array, 0, array.size)\n\n /**\n * Creates a byte array of
the specified [size], filled with random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n
*/\n public open
fun nextBytes(size: Int): ByteArray = nextBytes(ByteArray(size))\n\n /**\n * The default random
number generator.\n *\n * On JVM this generator is thread-safe, its methods can be invoked from multiple
threads.\n *\n * @sample samples.random.Randoms.defaultRandom\n
*/\n companion object Default :
Random(), Serializable {\n private val defaultRandom: Random = defaultPlatformRandom()\n private
object Serialized : Serializable {\n private const
val serialVersionUID = 0L\n private fun readResolve(): Any = Random\n }\n private fun
writeReplace(): Any = Serialized\n override fun nextBits(bitCount: Int): Int =
defaultRandom.nextBits(bitCount)\n override fun nextInt(): Int = defaultRandom.nextInt()\n override fun
nextInt(until: Int): Int = defaultRandom.nextInt(until)\n override fun nextInt(from: Int, until: Int): Int =
defaultRandom.nextInt(from, until)\n override fun nextLong(): Long = defaultRandom.nextLong()\n override
fun nextLong(until: Long): Long = defaultRandom.nextLong(until)\n override fun nextLong(from:
Long, until: Long): Long = defaultRandom.nextLong(from, until)\n override fun nextBoolean(): Boolean =
defaultRandom.nextBoolean()\n override fun nextDouble(): Double = defaultRandom.nextDouble()\n override
fun nextDouble(until: Double): Double = defaultRandom.nextDouble(until)\n override fun
nextDouble(from:
Double, until: Double): Double = defaultRandom.nextDouble(from, until)\n override fun nextFloat(): Float =
defaultRandom.nextFloat()\n override fun nextBytes(array: ByteArray): ByteArray =
defaultRandom.nextBytes(array)\n override fun nextBytes(size: Int): ByteArray =
defaultRandom.nextBytes(size)\n override fun nextBytes(array: ByteArray, fromIndex: Int, toIndex: Int):
ByteArray =\n defaultRandom.nextBytes(array, fromIndex, toIndex)\n }\n}\n\n/**\n * Returns a repeatable
random number generator seeded with the given [seed] `Int` value.\n *\n * Two generators with the same seed

```



```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextInt(from: UInt, until: UInt): UInt {\n checkUIntRangeBounds(from, until)\n\n val signedFrom =
from.toInt() xor Int.MIN_VALUE\n val signedUntil = until.toInt() xor Int.MIN_VALUE\n\n val signedResult =
nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE\n return signedResult.toUInt()\n}\n\n/**\n * Gets the next
random [UInt] from the random number generator in the specified [range].\n * Generates a [UInt] random value
uniformly distributed
in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n * @throws
IllegalArgumentException if [range] is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextInt(range: UIntRange): UInt = when {\n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n range.last < UInt.MAX_VALUE ->
nextInt(range.first, range.last + 1u)\n range.first > UInt.MIN_VALUE -> nextInt(range.first - 1u, range.last) +
1u\n else -> nextUInt()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator.\n * Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and
[ULong.MAX_VALUE] (inclusive).\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(): ULong = nextLong().toULong()\n\n/**\n * Gets the next random [ULong] from the random
number generator
less than the specified [until] bound.\n * Generates a [ULong] random value uniformly distributed between `0`
(inclusive) and the specified [until] bound (exclusive).\n * @throws IllegalArgumentException if [until] is
zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(until: ULong): ULong = nextULong(0uL, until)\n\n/**\n * Gets the next random [ULong] from
the random number generator in the specified range.\n * Generates a [ULong] random value uniformly
distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n * @throws
IllegalArgumentException if [from] is greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(from: ULong, until: ULong): ULong {\n checkULongRangeBounds(from, until)\n\n val
signedFrom = from.toLong() xor Long.MIN_VALUE\n val signedUntil = until.toLong() xor
Long.MIN_VALUE\n\n val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE\n return
signedResult.toULong()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator in the
specified [range].\n * Generates a [ULong] random value uniformly distributed in the specified [range]:\n * from
`range.start` inclusive to `range.endInclusive` inclusive.\n * @throws IllegalArgumentException if [range] is
empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(range: ULongRange): ULong = when {\n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n range.last < ULong.MAX_VALUE -
> nextULong(range.first, range.last + 1u)\n range.first > ULong.MIN_VALUE -> nextULong(range.first - 1u,
range.last) + 1u\n else -> nextULong()\n}\n\n/**\n * Fills the specified unsigned byte [array] with random bytes
and returns it.\n * @return [array] filled with
random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Random.nextUBytes(array: UByteArray): UByteArray {\n nextBytes(array.asByteArray())\n return
array\n}\n\n/**\n * Creates an unsigned byte array of the specified [size], filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(size: Int): UByteArray
= nextBytes(size).asUByteArray()\n\n/**\n * Fills a subrange of the specified `UByte` [array] starting from
[fromIndex] inclusive and ending [toIndex] exclusive with random UBytes.\n * @return [array] with the
subrange filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Random.nextUBytes(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {\n
 nextBytes(array.asByteArray(), fromIndex, toIndex)\n return array\n}\n\ninternal fun

```

```

checkUIntRangeBounds(from: UInt, until: UInt) = require(until > from) { boundsErrorMessage(from,
until) }\ninternal fun checkULongRangeBounds(from: ULong, until: ULong) = require(until > from) {
boundsErrorMessage(from, until) }\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\n/**\n * Random number generator, using Marsaglia's
\"xorwow\" algorithm\n */\n * Cycles after 2^192 - 2^32 repetitions.\n */\n * For more details, see Marsaglia, George
(July 2003). \"Xorshift RNGs\". Journal of Statistical Software. 8 (14). doi:10.18637/jss.v008.i14\n */\n * Available
at https://www.jstatsoft.org/v08/i14/paper\n */\n\ninternal class XorWowRandom internal constructor(\n private
var x: Int,\n private var y: Int,\n private var z: Int,\n private var w: Int,\n private var v: Int,\n private var
addend: Int)\n : Random(), Serializable {\n\n internal constructor(seed1: Int, seed2: Int) :\n
this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr 4))\n\n init {\n require((x or y or z
or w or v) != 0) { \"Initial state must have at least one non-zero element.\" }\n\n // some trivial seeds can
produce several values with zeroes in upper bits, so we discard first 64\n repeat(64) { nextInt() }\n }\n\n
override fun nextInt(): Int {\n // Equivalent to the xorwow algorithm\n // From Marsaglia, G. 2003. Xorshift
RNGs. J. Statis. Soft. 8, 14, p. 5\n var t = x\n t = t xor (t ushr 2)\n x = y\n y = z\n z = w\n
val v0 = v\n w = v0\n t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)\n v = t\n addend += 362437\n
return t + addend\n }\n\n override fun nextBits(bitCount: Int): Int =\n nextInt().takeUpperBits(bitCount)\n\n
private companion object {\n private const val serialVersionUID: Long = 0L\n }\n}\n", "/*\n * Copyright
2010-2023

```

```

JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin.ranges\n\n/**\n * An iterator over a progression of values of type `Char`.\n * @property
step the number by which the value is incremented on each step.\n */\n\ninternal class CharProgressionIterator(first:
Char, last: Char, val step: Int) : CharIterator() {\n private val finalElement: Int = last.code\n private var
hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Int = if (hasNext) first.code
else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun nextChar(): Char {\n val value
= next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n
hasNext = false\n }\n else {\n next
+= step\n }\n return value.toChar()\n }\n}\n\n/**\n * An iterator over a progression of values of type
`Int`.\n * @property step the number by which the value is incremented on each step.\n */\n\ninternal class
IntProgressionIterator(first: Int, last: Int, val step: Int) : IntIterator() {\n private val finalElement: Int = last\n
private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Int = if (hasNext) first
else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun nextInt(): Int {\n val
value = next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n
hasNext = false\n }\n else {\n next
+= step\n }\n return value\n }\n}\n\n/**\n * An
iterator over a progression of values of type `Long`.\n * @property step the number by which the value is
incremented on each step.\n */\n\ninternal class
LongProgressionIterator(first: Long, last: Long, val step: Long) : LongIterator() {\n private val finalElement:
Long = last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Long
= if (hasNext) first else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun
nextLong(): Long {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw
kotlin.NoSuchElementException()\n hasNext = false\n }\n else {\n next
+= step\n }\n return value\n }\n}\n}\n", "/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport
kotlin.internal.getProgressionLastElement\n\n/**\n * A progression of values of type
`Char`.\n */\n\npublic open class CharProgression\n internal constructor(\n (start: Char,\n
endInclusive: Char,\n step: Int\n) : Iterable<Char> {\n init {\n if (step == 0) throw

```

```

kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step == Int.MIN_VALUE) throw
kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on negation.")\n
}\n\n /**\n * The first element in the progression.\n */\n public val first: Char = start\n\n /**\n * The
last element in the progression.\n */\n public val last: Char = getProgressionLastElement(start.code,
endInclusive.code, step).toChar()\n\n /**\n * The step of the progression.\n */\n public val step: Int =
step\n\n override fun iterator(): CharIterator = CharProgressionIterator(first, last, step)\n\n /**\n * Checks if
the progression is empty.\n */\n * Progression with a positive step
is empty if its first element is greater than the last element.\n * Progression with a negative step is empty if its
first element is less than the last element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last
else first < last\n\n override fun equals(other: Any?): Boolean =\n other is CharProgression && (isEmpty()
&& other.isEmpty()) ||\n first == other.first && last == other.last && step == other.step)\n\n override fun
hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first.code + last.code) + step)\n\n override fun toString():
String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"\n\n companion object {\n
/**\n * Creates CharProgression within the specified bounds of a closed range.\n */\n * The
progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the
specified [step].\n * In order
to go backwards the [step] must be negative.\n */\n * [step] must be greater than `Int.MIN_VALUE` and
not equal to zero.\n */\n public fun fromClosedRange(rangeStart: Char, rangeEnd: Char, step: Int):
CharProgression = CharProgression(rangeStart, rangeEnd, step)\n } }\n\n /**\n * A progression of values of type
`Int`.\n */\n public open class IntProgression\n internal constructor\n (\n start: Int,\n endInclusive:
Int,\n step: Int\n): Iterable<Int> {\n init {\n if (step == 0) throw
kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step == Int.MIN_VALUE) throw
kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on negation.")\n
}\n\n /**\n * The first element in the progression.\n */\n public val first: Int = start\n\n /**\n * The last
element in the progression.\n */\n public val last: Int = getProgressionLastElement(start,
endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public val step: Int = step\n\n override
fun iterator(): IntIterator = IntProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is
empty.\n */\n * Progression with a positive step is empty if its first element is greater than the last element.\n
* Progression with a negative step is empty if its first element is less than the last element.\n */\n public open
fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n
 other is IntProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last &&
step == other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first + last) +
step)\n\n override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-
step}"\n\n
companion object {\n /**\n * Creates IntProgression within the specified bounds of a closed range.\n
*/\n * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it,
with the specified [step].\n * In order to go backwards the [step] must be negative.\n */\n * [step] must
be greater than `Int.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart:
Int, rangeEnd: Int, step: Int): IntProgression = IntProgression(rangeStart, rangeEnd, step)\n } }\n\n /**\n * A
progression of values of type `Long`.\n */\n public open class LongProgression\n internal constructor\n (\n
start: Long,\n endInclusive: Long,\n step: Long\n): Iterable<Long> {\n init {\n if (step ==
0L) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step == Long.MIN_VALUE)
throw kotlin.IllegalArgumentException("Step
must be greater than Long.MIN_VALUE to avoid overflow on negation.")\n }\n\n /**\n * The first element
in the progression.\n */\n public val first: Long = start\n\n /**\n * The last element in the progression.\n
*/\n */\n public val last: Long = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the
progression.\n */\n public val step: Long = step\n\n override fun iterator(): LongIterator =
LongProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n */\n *

```



```

@sample samples.ranges.Ranges.rangeFromComparable\n *\npublic operator fun <T : Comparable<T>>
T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)\n\n/**\n * Represents a range of [Comparable]
values.\n *\n@OptIn(ExperimentalStdlibApi::class)\nprivate open class ComparableOpenEndRange<T :
Comparable<T>>(\n
 override val start: T,\n override val endExclusive: T)\n) : OpenEndRange<T> {\n\n override fun equals(other:
Any?): Boolean {\n return other is ComparableOpenEndRange<*> && (isEmpty() && other.isEmpty() ||\n
 start == other.start && endExclusive == other.endExclusive)\n }\n\n override fun hashCode(): Int {\n
return if (isEmpty()) -1 else 31 * start.hashCode() + endExclusive.hashCode()\n }\n\n override fun toString():
String = \"\$start..\$endExclusive\"\n}\n\n/**\n * Creates an open-ended range from this [Comparable] value to the
specified [that] value.\n *\n * This value needs to be smaller than [that] value, otherwise the returned range will be
empty.\n *\n@sample samples.ranges.Ranges.rangeFromComparable\n
*\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun <T : Comparable<T>>
T.rangeUntil(that: T): OpenEndRange<T> = ComparableOpenEndRange(this, that)\n\n/**\n * Represents a range
of floating point numbers.\n *\n * Extends
[ClosedRange] interface providing custom operation [lessThanOrEquals] for comparing values of range domain
type.\n *\n * This interface is implemented by floating point ranges returned by [Float.rangeTo] and
[Double.rangeTo] operators to\n * achieve IEEE-754 comparison order instead of total order of floating point
numbers.\n *\n@SinceKotlin(\"1.1\")\npublic interface ClosedFloatingPointRange<T : Comparable<T>> :
ClosedRange<T> {\n override fun contains(value: T): Boolean = lessThanOrEquals(start, value) &&
lessThanOrEquals(value, endInclusive)\n override fun isEmpty(): Boolean = !lessThanOrEquals(start,
endInclusive)\n\n /**\n * Compares two values of range domain type and returns true if first is less than or
equal to second.\n *\n * fun lessThanOrEquals(a: T, b: T): Boolean\n }\n\n/**\n * A closed range of values of
type `Double`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n *\nprivate class
ClosedDoubleRange(\n start:
 Double,\n endInclusive: Double)\n) : ClosedFloatingPointRange<Double> {\n private val _start = start\n
private val _endInclusive = endInclusive\n override val start: Double get() = _start\n override val endInclusive:
Double get() = _endInclusive\n\n override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n\n
override fun contains(value: Double): Boolean = value >= _start && value <= _endInclusive\n override fun
isEmpty(): Boolean = !(_start <= _endInclusive)\n\n override fun equals(other: Any?): Boolean {\n return
other is ClosedDoubleRange && (isEmpty() && other.isEmpty() ||\n
 _start == other._start &&
 _endInclusive == other._endInclusive)\n }\n\n override fun hashCode(): Int {\n return if (isEmpty()) -1 else
31 * _start.hashCode() + _endInclusive.hashCode()\n }\n\n override fun toString(): String =
\"\$_start..$_endInclusive\"\n}\n\n/**\n * Creates a range from this [Double] value to the specified [that]
value.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n *\n@sample
samples.ranges.Ranges.rangeFromDouble\n *\n@SinceKotlin(\"1.1\")\npublic operator fun Double.rangeTo(that:
Double): ClosedFloatingPointRange<Double> = ClosedDoubleRange(this, that)\n\n/**\n * An open-ended range of
values of type `Double`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n
*\n@OptIn(ExperimentalStdlibApi::class)\nprivate class OpenEndDoubleRange(\n start: Double,\n
endExclusive: Double)\n) : OpenEndRange<Double> {\n private val _start = start\n private val _endExclusive =
endExclusive\n override val start: Double get() = _start\n override val endExclusive: Double get() =
_endExclusive\n\n private fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n\n override fun
contains(value: Double): Boolean = value >= _start && value < _endExclusive\n override fun isEmpty(): Boolean
= !(_start < _endExclusive)\n\n
 override fun equals(other: Any?): Boolean {\n return other is OpenEndDoubleRange && (isEmpty() &&
other.isEmpty() ||\n
 _start == other._start && _endExclusive == other._endExclusive)\n }\n\n override
fun hashCode(): Int {\n return if (isEmpty()) -1 else 31 * _start.hashCode() + _endExclusive.hashCode()\n
 }\n\n override fun toString(): String = \"\$_start..$_endExclusive\"\n}\n\n/**\n * Creates an open-ended range
from this [Double] value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range

```

according to IEEE-754.

```

@SinceKotlin("1.7")
@ExperimentalStdlibApi
public operator fun
Double.rangeUntil(that: Double): OpenEndRange<Double> = OpenEndDoubleRange(this, that)
A
closed range of values of type `Float`.
Numbers are compared with the ends of this range according to IEEE-
754.
private class ClosedFloatRange(
 start: Float,
 endInclusive: Float
):
ClosedFloatingPointRange<Float>
{
 private val _start = start
 private val _endInclusive = endInclusive
 override val start: Float get() =
_start
 override val endInclusive: Float get() = _endInclusive
 override fun lessThanOrEquals(a: Float, b:
Float): Boolean = a <= b
 override fun contains(value: Float): Boolean = value >= _start && value <=
_endInclusive
 override fun isEmpty(): Boolean = !(_start <= _endInclusive)
 override fun equals(other:
Any?): Boolean {
 return other is ClosedFloatRange && (isEmpty() && other.isEmpty()) ||
_start
== other._start && _endInclusive == other._endInclusive
 }
 override fun hashCode(): Int {
 return if
(isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()
 }
 override fun toString(): String =
"\$_start..$_endInclusive"
}
Creates a range from this [Float] value to the specified [that] value.
Numbers are compared with the ends of this range according
to IEEE-754.
@sample samples.ranges.Ranges.rangeFromFloat
@SinceKotlin("1.1")
public operator
fun Float.rangeTo(that: Float): ClosedFloatingPointRange<Float> = ClosedFloatRange(this, that)
An
open-ended range of values of type `Float`.
Numbers are compared with the ends of this range according to
IEEE-754.
@OptIn(ExperimentalStdlibApi::class)
private class OpenEndFloatRange(
 start: Float,
 endExclusive: Float
): OpenEndRange<Float> {
 private val _start = start
 private val _endExclusive =
endExclusive
 override val start: Float get() = _start
 override val endExclusive: Float get() =
_endExclusive
 private fun lessThanOrEquals(a: Float, b: Float): Boolean = a <= b
 override fun
contains(value: Float): Boolean = value >= _start && value < _endExclusive
 override fun isEmpty(): Boolean =
!(_start < _endExclusive)
 override fun equals(other: Any?): Boolean {
 return other is
OpenEndFloatRange
&& (isEmpty() && other.isEmpty()) ||
_start == other._start && _endExclusive ==
other._endExclusive
 }
 override fun hashCode(): Int {
 return if (isEmpty()) -1 else 31 *
_start.hashCode() + _endExclusive.hashCode()
 }
 override fun toString(): String =
"\$_start..<$_endExclusive"
}
Creates an open-ended range from this [Float] value to the specified
[that] value.
Numbers are compared with the ends of this range according to IEEE-754.
@SinceKotlin("1.7")
@ExperimentalStdlibApi
public operator fun Float.rangeUntil(that: Float):
OpenEndRange<Float> = OpenEndFloatRange(this, that)
Returns `true` if this iterable range contains
the specified [element].
Always returns `false` if the [element] is `null`.
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline operator fun <T, R> R.contains(element: T?):
Boolean where T : Any, R : ClosedRange<T>, R : Iterable<T> =
element
!= null && contains(element)
Returns `true` if this iterable range contains the specified [element].
Always returns `false` if the [element] is `null`.
@SinceKotlin("1.7")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline operator fun <T,
R> R.contains(element: T?): Boolean where T : Any, R : OpenEndRange<T>, R : Iterable<T> =
element != null
&& contains(element)
internal fun checkStepIsPositive(isPositive: Boolean, step: Number) {
 if (!isPositive)
throw IllegalArgumentException("Step must be positive, was: $step.")
}
Copyright 2010-2019
JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.
@file:kotlin.jvm.JvmName("KClasses")
@file:Suppress("UNCHECKED_CAST")
package
kotlin.reflect
nimport kotlin.internal.LowPriorityInOverloadResolution
Casts the given [value]
to the class represented by this [KClass] object.
Throws an exception if the value is `null` or if it is not an
instance of this class.
This is an experimental function that behaves as a similar function from
kotlin.reflect.full on JVM.
@see [KClass.isInstance]
@see [KClass.safeCast]
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@LowPriorityInOverloadResoluti

```



on fun <T : Any> KClass<T>.cast(value: Any?): T {  
 if (!isInstance(value)) throw ClassCastException("Value cannot be cast to \$qualifiedOrSimpleName")  
 return value as T  
}  
// TODO: replace with qualifiedName when it is fully supported in K/JS  
internal expect val KClass<\*>.qualifiedOrSimpleName: String?  
Casts the given [value] to the class represented by this [KClass] object.  
Returns `null` if the value is `null` or if it is not an instance of this class.  
This is an experimental function that behaves as a similar function from `kotlin.reflect.full`

on JVM.  
@see [KClass.isInstance]  
@see [KClass.cast]  
@SinceKotlin("1.4")  
@WasExperimental(ExperimentalStdlibApi::class)  
@LowPriorityInOverloadResolution  
on fun <T : Any> KClass<T>.safeCast(value: Any?): T? {  
 return if (isInstance(value)) value as T else null  
}  
Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.  
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  
package kotlin.reflect  
import kotlin.jvm.JvmField  
import kotlin.jvm.JvmStatic  
Represents a type projection. Type projection is usually the argument to another type in a type usage.  
For example, in the type `Array<out Number>`, `out Number` is the covariant projection of the type represented by the class `Number`.  
Type projection is either the star projection, or an entity consisting of a specific type plus optional variance.  
See the [Kotlin

language documentation](https://kotlinlang.org/docs/reference/generics.html#type-projections) for more information.  
@SinceKotlin("1.1")  
public data class KTypeProjection constructor(  
 /\*\*  
 \* The use-site variance specified in the projection, or `null` if this is a star projection.  
 \*/  
 public val variance: KVariance?,  
 /\*\*  
 \* The type specified in the projection, or `null` if this is a star projection.  
 \*/  
 public val type: KType?) {  
 init {  
 require((variance == null) == (type == null)) {  
 if (variance == null)  
 "Star projection must have no type specified."  
 else  
 "The projection variance \$variance requires type to be specified."  
 }  
 }  
 override fun toString(): String = when (variance) {  
 null -> "/\*"  
 KVariance.INVARIANT -> type.toString()  
 KVariance.IN -> "in \$type"  
 KVariance.OUT -> "out \$type"  
 }  
}

public companion object {  
 // provided for compiler access  
 @JvmField  
 @PublishedApi  
 internal val star: KTypeProjection = KTypeProjection(null, null)  
 /\*\*  
 \* Star projection, denoted by the `\*` character.  
 \* For example, in the type `KClass<\*>`, `\*` is the star projection.  
 \* See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#star-projections) for more information.  
 \*/  
 public val STAR: KTypeProjection get() = star  
 /\*\*  
 \* Creates an invariant projection of a given type. Invariant projection is just the type itself,  
 \* without any use-site variance modifiers applied to it.  
 \* For example, in the type `Set<String>`, `String` is an invariant projection of the type represented by the class `String`.  
 \*/  
 @JvmStatic  
 public fun invariant(type: KType):

KTypeProjection =  
 KTypeProjection(KVariance.INVARIANT, type)  
 /\*\*  
 \* Creates a contravariant projection of a given type, denoted by the `in` modifier applied to a type.  
 \* For example, in the type `MutableList<in Number>`, `in Number` is a contravariant projection of the type of class `Number`.  
 \*/  
 @JvmStatic  
 public fun contravariant(type: KType):

KTypeProjection =  
 KTypeProjection(KVariance.IN, type)  
 /\*\*  
 \* Creates a covariant projection of a given type, denoted by the `out` modifier applied to a type.  
 \* For example, in the type `Array<out Number>`, `out Number` is a covariant projection of the type of class `Number`.  
 \*/  
 @JvmStatic  
 public fun covariant(type: KType): KTypeProjection =

KTypeProjection(KVariance.OUT, type)  
}  
Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.  
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  
package kotlin.reflect  
Represents variance applied to a type parameter on the declaration site (\*declaration-site variance\*), or to a type in a projection (\*use-site variance\*).  
See the [Kotlin language

documentation](https://kotlinlang.org/docs/reference/generics.html#variance) for more information.  
@see [KTypeParameter.variance]  
@see [KTypeProjection]  
@SinceKotlin("1.1")  
enum class KVariance



```

Appendable = append(value).appendLine()\n\n\ninternal fun <T> Appendable.appendElement(element: T,
transform: ((T) -> CharSequence)?) {\n when {\n transform != null -> append(transform(element))\n element is CharSequence? -> append(element)\n element is Char -> append(element)\n else ->
append(element.toString())\n }\n}\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\n/**\n * Trims leading whitespace characters followed by [marginPrefix] from every line of a source
string and removes\n * the first and the last lines if they
are blank (notice difference blank vs empty).\n *\n * Doesn't affect a line if it doesn't contain [marginPrefix] except
the first and the last blank lines.\n *\n * Doesn't preserve the original line endings.\n *\n * @param marginPrefix
non-blank string, which is used as a margin delimiter. Default is `|` (pipe character).\n *\n * @sample
samples.text.Strings.trimMargin\n * @see trimIndent\n * @see kotlin.text.isWhitespace\n
*/\n\n@kotlin.internal.IntrinsicConstEvaluation\n\npublic fun String.trimMargin(marginPrefix: String = "|"): String
=\n replaceIndentByMargin("|", marginPrefix)\n\n/**\n * Detects indent by [marginPrefix] as it does
[trimMargin] and replace it with [newIndent].\n *\n * @param marginPrefix non-blank string, which is used as a
margin delimiter. Default is `|` (pipe character).\n *\n * @public fun String.replaceIndentByMargin(newIndent: String =
"|", marginPrefix: String = "|"): String {\n require(marginPrefix.isNotBlank()) { `marginPrefix must be non-
blank string.`\n
}\n val lines = lines()\n return lines.reindent(length + newIndent.length * lines.size,
getIndentFunction(newIndent), { line ->\n val firstNonWhitespaceIndex = line.indexOfFirst { !it.isWhitespace()
}\n when {\n firstNonWhitespaceIndex == -1 -> null\n line.startsWith(marginPrefix,
firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex + marginPrefix.length)\n else -> null\n
}\n })\n}\n\n/**\n * Detects a common minimal indent of all the input lines, removes it from every line and
also removes the first and the last\n * lines if they are blank (notice difference blank vs empty).\n *\n * Note that
blank lines do not affect the detected indent level.\n *\n * In case if there are non-blank lines with no leading
whitespace characters (no indent at all) then the\n * common indent is 0, and therefore this function doesn't change
the indentation.\n *\n * Doesn't preserve the original line endings.\n *\n * @sample
samples.text.Strings.trimIndent\n * @see trimMargin\n * @see kotlin.text.isBlank\n
*/\n\n@kotlin.internal.IntrinsicConstEvaluation\n\npublic fun String.trimIndent(): String = replaceIndent("\\")\n\n/**\n * Detects a common minimal indent like it does [trimIndent] and replaces it with the specified [newIndent].\n
*/\n\npublic fun String.replaceIndent(newIndent: String = "\\"): String {\n val lines = lines()\n val
minCommonIndent = lines\n .filter(String::isNotBlank)\n .map(String::indentWidth)\n .minOrNull() ?:
0\n return lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), { line ->
line.drop(minCommonIndent) })\n}\n\n/**\n * Prepends [indent] to every line of the original string.\n *\n * Doesn't
preserve the original line endings.\n *\n * @public fun String.prependIndent(indent: String = " "): String =\nlineSequence()\n .map {\n when {\n it.isBlank() -> {\n when {\n
 it.length < indent.length -> indent\n else -> it\n }\n }\n else -> indent + it\n }\n }.joinToString("\\n")\n\nprivate fun String.indentWidth(): Int =
indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }\n\nprivate fun getIndentFunction(indent: String)
= when {\n indent.isEmpty() -> { line: String -> line }\n else -> { line: String -> indent + line }\n}\n\nprivate
inline fun List<String>.reindent(\n resultSizeEstimate: Int,\n indentAddFunction: (String) -> String,\n indentCutFunction: (String) -> String?): String {\n val lastIndex = lastIndex\n return mapIndexedNotNull {
index, value ->\n if ((index == 0 || index == lastIndex) && value.isBlank())\n null\n else\n indentCutFunction(value)?.let(indentAddFunction)?: value\n }.joinTo(StringBuilder(resultSizeEstimate),
"\n")\n
.toString()\n}\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

```

```

license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Defines names for Unicode symbols used in proper
Typography.\n */\npublic object Typography {\n /** The character " \u2013 quotation mark */\n public
const val quote: Char = "\u0022"\n /** The character $ \u2013 dollar sign */\n public const val dollar:
Char = "\u0024"\n /** The character & \u2013 ampersand */\n public const val amp: Char = "\u0026"\n
 /** The character < \u2013 less-than sign */\n public const val less: Char = "\u003C"\n /** The character
> \u2013 greater-than sign */\n public const val greater: Char = "\u003E"\n /** The non-breaking space
character */\n public const val nbsp: Char = "\u00A0"\n /** The character × */\n public
const val times: Char = "\u00D7"\n /** The character ¢ */\n public const val cent: Char = "\u00A2"\n
 /** The character £ */\n public const val pound: Char = "\u00A3"\n /** The character § */\n
 public const val section: Char = "\u00A7"\n /** The character © */\n public const val copyright: Char =
"\u00A9"\n /** The character « */\n @SinceKotlin("1.6")\n public const val leftGuillemet: Char =
"\u00AB"\n /** The character » */\n @SinceKotlin("1.6")\n public const val rightGuillemet: Char =
"\u00BB"\n /** The character ® */\n public const val registered: Char = "\u00AE"\n /** The character
° */\n public const val degree: Char = "\u00B0"\n /** The character ± */\n public const val
plusMinus: Char = "\u00B1"\n /** The character ¶ */\n public const val paragraph: Char = "\u00B6"\n
 /** The character · */\n public const val middleDot: Char = "\u00B7"\n
 /** The character ½ */\n public const val half: Char = "\u00BD"\n /** The character – */\n
 public const val ndash: Char = "\u2013"\n /** The character — */\n public const val mdash: Char =
"\u2014"\n /** The character ‘ */\n public const val leftSingleQuote: Char = "\u2018"\n /** The
character ’ */\n public const val rightSingleQuote: Char = "\u2019"\n /** The character ‚ */\n
 public const val lowSingleQuote: Char = "\u201A"\n /** The character “ */\n public const val
leftDoubleQuote: Char = "\u201C"\n /** The character ” */\n public const val rightDoubleQuote: Char
= "\u201D"\n /** The character „ */\n public const val lowDoubleQuote: Char = "\u201E"\n /** The
character † */\n public const val dagger: Char = "\u2020"\n /** The character ‡ */\n public
const val doubleDagger: Char = "\u2021"\n /** The character • */\n public
const val bullet: Char = "\u2022"\n /** The character … */\n public const val ellipsis: Char = "\u2026"\n
 /** The character ′ */\n public const val prime: Char = "\u2032"\n /** The character ″ */\n
 public const val doublePrime: Char = "\u2033"\n /** The character € */\n public const val euro: Char =
"\u20AC"\n /** The character ™ */\n public const val tm: Char = "\u2122"\n /** The character
≈ */\n public const val almostEqual: Char = "\u2248"\n /** The character ≠ */\n public const
val notEqual: Char = "\u2260"\n /** The character ≤ */\n public const val lessOrEqual: Char =
"\u2264"\n /** The character ≥ */\n public const val greaterOrEqual: Char = "\u2265"\n\n /** The
character « */\n @Deprecated("This constant has a typo in the name. Use leftGuillemet instead.",
ReplaceWith("Typography.leftGuillemet"))\n @DeprecatedSinceKotlin("1.6")\n public const val leftGuillemete: Char = "\u00AB"\n\n /** The character » */\n @Deprecated("This
constant has a typo in the name. Use rightGuillemet instead.", ReplaceWith("Typography.rightGuillemet"))\n @DeprecatedSinceKotlin("1.6")\n public const val rightGuillemete: Char = "\u00BB"\n}"/*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Represents a collection of captured groups in a single match of a regular expression.\n */\n * This collection has size
of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n */\n * An element of the
collection at the particular index can be `null`,\n * if the corresponding group in the regular
expression is optional and\n * there was no match captured by that group.\n */\n\npublic interface
MatchGroupCollection : Collection<MatchGroup?> {\n /** Returns a group with the specified [index].\n */\n * @return An instance of [MatchGroup] if the group with the specified [index] was matched or `null` otherwise.\n
 */\n * Groups are indexed from 1 to the count of groups in the regular expression. A group with the index 0\n *
corresponds to the entire match.\n */\n public operator fun get(index: Int): MatchGroup?\n}\n\n/**\n * Extends

```

```

[MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.\n
*\n@SinceKotlin("1.1")\npublic interface MatchNamedGroupCollection : MatchGroupCollection {\n /**\n *
Returns a named group with the specified [name].\n * @return An instance of [MatchGroup] if the group with the
specified [name] was matched or `null` otherwise.\n * @throws IllegalArgumentException if there is
no group with the specified [name] defined in the regex pattern.\n * @throws UnsupportedOperationException if
this match group collection doesn't support getting match groups by name.\n * for example, when it's not
supported by the current platform.\n */\n public operator fun get(name: String): MatchGroup?\n}\n/**\n *
Represents the results from a single regular expression match.\n */\npublic interface MatchResult {\n /** The
range of indices in the original string where match was captured. *\n public val range: IntRange\n /** The
substring from the input string captured by this match. *\n public val value: String\n /**\n * A collection of
groups matched by the regular expression.\n *\n * This collection has size of `groupCount + 1` where
`groupCount` is the count of groups in the regular expression.\n * Groups are indexed from 1 to `groupCount` and
group with the index 0 corresponds to the entire match.\n */\n public val groups:
MatchGroupCollection\n /**\n * A list of matched indexed group values.\n *\n * This list has size of
`groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n *\n * If the group in the
regular expression is optional and there were no match captured by that group,\n * corresponding item in
[groupValues] is an empty string.\n *\n * @sample
samples.text.Regexps.matchDestructuringToGroupValues\n */\n public val groupValues: List<String>\n}\n
/**\n * An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment
of group values.\n *\n * component1 corresponds to the value of the first group, component2 \u2014 of the
second, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n
public val destructured: Destructured
get() = Destructured(this)\n\n /** Returns a new [MatchResult] with the results for the next match, starting at the
position\n * at which the last match ended (at the character after the last matched character).\n */\n public fun
next(): MatchResult?\n\n /**\n * Provides components for destructuring assignment of group values.\n *\n
* [component1] corresponds to the value of the first group, [component2] \u2014 of the second, and so on.\n *\n
* If the group in the regular expression is optional and there were no match captured by that group,\n *
corresponding component value is an empty string.\n *\n * @sample
samples.text.Regexps.matchDestructuringToGroupValues\n */\n public class Destructured internal
constructor(public val match: MatchResult) {\n @kotlin.internal.InlineOnly\n public operator inline fun
component1(): String = match.groupValues[1]\n @kotlin.internal.InlineOnly\n public operator inline
fun component2(): String = match.groupValues[2]\n @kotlin.internal.InlineOnly\n public operator inline
fun component3(): String = match.groupValues[3]\n @kotlin.internal.InlineOnly\n public operator inline
fun component4(): String = match.groupValues[4]\n @kotlin.internal.InlineOnly\n public operator inline
fun component5(): String = match.groupValues[5]\n @kotlin.internal.InlineOnly\n public operator inline
fun component6(): String = match.groupValues[6]\n @kotlin.internal.InlineOnly\n public operator inline
fun component7(): String = match.groupValues[7]\n @kotlin.internal.InlineOnly\n public operator inline
fun component8(): String = match.groupValues[8]\n @kotlin.internal.InlineOnly\n public operator inline
fun component9(): String = match.groupValues[9]\n @kotlin.internal.InlineOnly\n public operator inline
fun component10(): String = match.groupValues[10]\n\n
/**\n * Returns destructured group values as a list of strings.\n * First value in the returned list
corresponds to the value of the first group, and so on.\n *\n * @sample
samples.text.Regexps.matchDestructuringToGroupValues\n */\n public fun toList(): List<String> =
match.groupValues.subList(1, match.groupValues.size)\n }\n}", "\n\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass()\n@file:kotlin.jvm.JvmName("DurationUnitKt")\n\npackage

```

```

kotlin.time\n\n\n/**\n * The list of possible time measurement units, in which a duration can be expressed.\n *\n *
The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24
[HOURLS].\n *\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic
expect enum class DurationUnit {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of a
microsecond.\n *\n * NANOSECONDS,\n */\n * Time unit representing one microsecond, which is 1/1000
of a millisecond.\n *\n * MICROSECONDS,\n */\n * Time unit representing one millisecond, which is
1/1000 of a second.\n *\n * MILLISECONDS,\n */\n * Time unit representing one second.\n *\n
SECONDS,\n */\n * Time unit representing one minute.\n *\n * MINUTES,\n */\n * Time unit
representing one hour.\n *\n * HOURS,\n */\n * Time unit representing one day, which is always equal to
24 hours.\n *\n * DAYS;\n }\n\n\n/** Converts the given time duration [value] expressed in the specified
[sourceUnit] into the specified [targetUnit]. *\n@SinceKotlin("1.3")\ninternal expect fun
convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double\n\n// overflown
result
is unspecified\n@SinceKotlin("1.5")\ninternal expect fun convertDurationUnitOverflow(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Long\n\n// overflown result is coerced in the Long range
boundaries\n@SinceKotlin("1.5")\ninternal expect fun convertDurationUnit(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit):
Long\n\n\n@SinceKotlin("1.3")\n@Suppress("REDUNDANT_ELSE_IN_WHEN")\ninternal fun
DurationUnit.shortName(): String = when (this) {\n DurationUnit.NANOSECONDS -> "\ns"\n
DurationUnit.MICROSECONDS -> "\nus"\n DurationUnit.MILLISECONDS -> "\ms"\n
DurationUnit.SECONDS -> "\s"\n DurationUnit.MINUTES -> "\m"\n DurationUnit.HOURS -> "\h"\n
DurationUnit.DAYS -> "\d"\n else -> error("Unknown unit: $this")\n }\n\n@SinceKotlin("1.5")\ninternal fun
durationUnitByShortName(shortName: String): DurationUnit = when (shortName) {\n "\ns" ->
DurationUnit.NANOSECONDS\n "\us" -> DurationUnit.MICROSECONDS\n "\ms"
-> DurationUnit.MILLISECONDS\n "\s" -> DurationUnit.SECONDS\n "\m" -> DurationUnit.MINUTES\n
"\h" -> DurationUnit.HOURS\n "\d" -> DurationUnit.DAYS\n else -> throw
IllegalArgumentException("Unknown duration unit short name:
$shortName")\n }\n\n@SinceKotlin("1.5")\ninternal fun durationUnitByIsoChar(isoChar: Char,
isTimeComponent: Boolean): DurationUnit =\n when {\n !isTimeComponent -> {\n when (isoChar)
{\n 'D' -> DurationUnit.DAYS\n else -> throw IllegalArgumentException("Invalid or
unsupported duration ISO non-time unit: $isoChar")\n }\n }\n else -> {\n when (isoChar) {\n
 'H' -> DurationUnit.HOURS\n 'M' -> DurationUnit.MINUTES\n 'S' ->
DurationUnit.SECONDS\n else -> throw IllegalArgumentException("Invalid duration ISO time unit:
$isoChar")\n }\n }\n }, /*\n * Copyright 2010-2019 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport
kotlin.annotation.AnnotationTarget.\n\n/**\n *\n * This annotation marks the experimental preview of the standard
library API for measuring time and working with durations.\n *\n * > Note that this API is in a preview state and has
a very high chance of being changed in the future.\n *\n * Do not use it if you develop a library since your library will
become binary incompatible\n *\n * with the future versions of the standard library.\n *\n * Any usage of a declaration
annotated with `@ExperimentalTime` must be accepted either by\n *\n * annotating that usage with the [OptIn]
annotation, e.g. `@OptIn(ExperimentalTime::class)`,\n *\n * or by using the compiler argument `--opt-in=kotlin.time.ExperimentalTime`. \n *\n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n TYPEALIAS)\n@SinceKotlin("1.3")\npublic annotation class
ExperimentalTime\n\n/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language

```

```

contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.jvm.JvmInline\n\n/**\n * A source of time
for measuring time intervals.\n *\n * The only operation provided by the time source is [markNow]. It returns a
[TimeMark], which can be used to query the elapsed time later.\n *\n * @see [measureTime]\n *\n * @see
[measureTimedValue]\n *\n * @SinceKotlin("1.3")\n * @ExperimentalTime\n * public interface TimeSource {\n */\n
 * Marks a point in time on this time source.\n *\n * The returned [TimeMark]
instance encapsulates the captured time point and allows querying\n *\n * the duration of time interval
[elapsed][TimeMark.elapsedNow] from that point.\n *\n * public fun markNow(): TimeMark\n\n */\n *\n * A
[TimeSource] that returns [time marks][ComparableTimeMark] that can be compared for difference with each
other.\n *\n * @SinceKotlin("1.8")\n * @ExperimentalTime\n * public interface WithComparableMarks :
TimeSource {\n *\n * override fun markNow(): ComparableTimeMark\n *\n *\n */\n *\n * The most precise time
source available in the platform.\n *\n * This time source returns its readings from a source of monotonic time
when it is available in a target platform,\n *\n * and resorts to a non-monotonic time source otherwise.\n *\n *
The function [markNow] of this time source returns the specialized [ValueTimeMark] that is an inline value class\n
 * wrapping a platform-dependent time reading value.\n *\n * public object Monotonic :
TimeSource.WithComparableMarks
{\n *\n * override fun markNow(): ValueTimeMark = MonotonicTimeSource.markNow()\n *\n * override fun
toString(): String = MonotonicTimeSource.toString()\n\n */\n *\n * A specialized [kotlin.time.TimeMark]
returned by [TimeSource.Monotonic] time source.\n *\n * This time mark is implemented as an inline
value class wrapping a platform-dependent\n *\n * time reading value of the default monotonic time source, thus
allowing to avoid additional boxing\n *\n * of that value.\n *\n * The operations [plus] and [minus] are
also specialized to return [ValueTimeMark] type.\n *\n * This time mark implements
[ComparableTimeMark] and therefore is comparable with other time marks\n *\n * obtained from the same
[TimeSource.Monotonic] time source.\n *\n * @ExperimentalTime\n * @SinceKotlin("1.7")\n
@JvmInline\n * public value class ValueTimeMark internal constructor(internal
val reading: ValueTimeMarkReading) : ComparableTimeMark {\n *\n * override fun elapsedNow(): Duration =
MonotonicTimeSource.elapsedFrom(this)\n *\n * override fun plus(duration: Duration): ValueTimeMark =
MonotonicTimeSource.adjustReading(this, duration)\n *\n * override fun minus(duration: Duration):
ValueTimeMark = MonotonicTimeSource.adjustReading(this, -duration)\n *\n * override fun hasPassedNow():
Boolean = !elapsedNow().isNegative()\n *\n * override fun hasNotPassedNow(): Boolean =
elapsedNow().isNegative()\n *\n * override fun minus(other: ComparableTimeMark): Duration {\n *\n * if
(other !is ValueTimeMark)\n *\n * throw IllegalArgumentException("Subtracting or comparing time marks
from different time sources is not possible: $this and $other")\n *\n * return this.minus(other)\n *\n *\n
*/\n *\n * Returns the duration elapsed between the [other] time mark obtained from the same
[TimeSource.Monotonic] time source and `this` time mark.\n *\n * The returned duration can be
infinite if the time marks are far away from each other and\n *\n * the result doesn't fit into [Duration] type,\n
 *\n * or if one time mark is infinitely distant, or if both `this` and [other] time marks\n *\n * lie infinitely distant
on the opposite sides of the time scale.\n *\n * Two infinitely distant time marks on the same side of
the time scale are considered equal and\n *\n * the duration between them is [Duration.ZERO].\n *\n *\n
 *\n * public operator fun minus(other: ValueTimeMark): Duration = MonotonicTimeSource.differenceBetween(this,
other)\n\n */\n *\n * Compares this time mark with the [other] time mark for order.\n *\n *\n
 * - Returns zero if this time mark represents *the same moment* of time as the [other] time mark.\n *\n * -
Returns
a negative number if this time mark is *earlier* than the [other] time mark.\n *\n * - Returns a positive number
if this time mark is *later* than the [other] time mark.\n *\n *\n * public operator fun compareTo(other:
ValueTimeMark): Int =\n *\n * (this - other).compareTo(Duration.ZERO)\n *\n *\n *\n * public companion
object {\n *\n * }\n\n */\n *\n * A platform-specific reading type that is wrapped by
[TimeSource.Monotonic.ValueTimeMark] inline class. *\n *\n * internal expect class ValueTimeMarkReading\n\n */\n

```

\* Represents a time point notched on a particular [TimeSource]. Remains bound to the time source it was taken from and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).

\* Returns the amount of time passed from this mark measured with the time source from which this mark was taken. Note that the value returned by this function can change on subsequent invocations.

@throws IllegalArgumentException an implementation may throw if calculating the elapsed time involves adding a positive infinite duration to an infinitely distant past time mark or a negative infinite duration to an infinitely distant future time mark.

public abstract fun elapsedNow(): Duration

Returns a time mark on the same time source that is ahead of this time mark by the specified [duration]. The returned time mark is more `_late_` when the [duration] is positive, and more `_early_` when the [duration] is negative. If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark. In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.

@throws IllegalArgumentException an implementation may throw if a positive infinite duration is added to an infinitely distant past time mark or a negative infinite duration is added to an infinitely distant future time mark.

public operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)

Returns a time mark on the same time source that is behind this time mark by the specified [duration]. The returned time mark is more `_early_` when the [duration] is positive, and more `_late_` when the [duration] is negative. If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark. In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.

@throws IllegalArgumentException an implementation may throw if a positive infinite duration is subtracted from an infinitely distant future time mark or a negative infinite duration is subtracted from an infinitely distant past time mark.

public open operator fun minus(duration: Duration): TimeMark = plus(-duration)

Returns true if this time mark has passed according to the time source from which this mark was taken. Note that the value returned by this function can change on subsequent invocations. If the time source is monotonic, it can change only from `false` to `true`, namely, when the time mark becomes behind the current point of the time source.

public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()

Returns false if this time mark has not passed according to the time source from which this mark was taken. Note that the value returned by this function can change on subsequent invocations. If the time source is monotonic, it can change only from `true` to `false`, namely, when the time mark becomes behind the current point of the time source.

public fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()

A [TimeMark] that can be compared for difference with other time marks obtained from the same [TimeSource.WithComparableMarks] time source.

@SinceKotlin("1.8")@ExperimentalTime\npublic interface ComparableTimeMark : TimeMark, Comparable<ComparableTimeMark> {\n public abstract override operator fun plus(duration: Duration): ComparableTimeMark\n public open override operator fun minus(duration: Duration): ComparableTimeMark = plus(-duration)\n /\*\*\n \* Returns the duration elapsed between the [other] time mark and `this` time mark.\n \* The returned duration can be infinite if the time marks are far away from each other and the result doesn't fit into [Duration] type, or if one time mark is infinitely distant, or if both `this` and [other] time marks lie infinitely distant on the opposite sides of the time scale.\n \* Two infinitely distant time marks on the same side of the time scale are considered equal and the duration between them is [Duration.ZERO].\n \* Note that the other time mark must be obtained from the same time source as this one.\n \* @throws IllegalArgumentException if time marks were obtained from different time sources.\n \*/\n public operator fun minus(other: ComparableTimeMark): Duration\n /\*\*\n \* Compares this time mark with the [other] time mark for order.\n \* - Returns zero if this time mark represents *the same moment* of time as the [other] time mark.\n \* - Returns a negative number if this time mark is *earlier* than the [other] time mark.\n \* - Returns a positive number if this time mark is *later* than the [other] time mark.\n \* Note



```

that the other time mark must be obtained from the same time source as this one.\n * \n * @throws
IllegalArgumentExpection if time marks were obtained from different
time sources.\n * \n public override operator fun compareTo(other: ComparableTimeMark): Int = \n (this -
other).compareTo(Duration.ZERO)\n\n /** \n * Returns `true` if two time marks from the same time source
represent the same moment of time, and `false` otherwise, \n * including the situation when the time marks were
obtained from different time sources.\n * \n override fun equals(other: Any?): Boolean \n override fun
hashCode(): Int \n} \n\n @ExperimentalTime \n private class AdjustedTimeMark(val mark: TimeMark, val
adjustment: Duration) : TimeMark { \n override fun elapsedNow(): Duration = mark.elapsedNow() -
adjustment \n\n override fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment +
duration) \n} \n\n /** \n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors. \n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. \n
*/ \n\n package kotlin.time \n\n import
kotlin.time.Duration.Companion.milliseconds \n\n import kotlin.time.Duration.Companion.nanoseconds \n\n import
kotlin.time.Duration.Companion.seconds \n\n\n @SinceKotlin("1.3") \n\n @ExperimentalTime \n\n internal expect object
MonotonicTimeSource : TimeSource.WithComparableMarks { \n override fun markNow():
TimeSource.Monotonic.ValueTimeMark \n fun elapsedFrom(timeMark:
TimeSource.Monotonic.ValueTimeMark): Duration \n fun differenceBetween(one:
TimeSource.Monotonic.ValueTimeMark, another: TimeSource.Monotonic.ValueTimeMark): Duration \n fun
adjustReading(timeMark: TimeSource.Monotonic.ValueTimeMark, duration: Duration):
TimeSource.Monotonic.ValueTimeMark \n} \n\n /** \n * An abstract class used to implement time sources that return
their readings as [Long] values in the specified [unit]. \n * \n * @property unit The unit in which this time source's
readings are expressed. \n * \n @SinceKotlin("1.3") \n\n @ExperimentalTime \n\n public abstract class
AbstractLongTimeSource(protected val unit:
DurationUnit) : TimeSource.WithComparableMarks { \n /** \n * This protected method should be overridden to
return the current reading of the time source expressed as a [Long] number \n * in the unit specified by the [unit]
property. \n * \n protected abstract fun read(): Long \n\n private class LongTimeMark(private val startedAt:
Long, private val timeSource: AbstractLongTimeSource, private val offset: Duration) : ComparableTimeMark { \n
override fun elapsedNow(): Duration = if (offset.isInfinite()) -offset else (timeSource.read() -
startedAt).toDuration(timeSource.unit) - offset \n override fun plus(duration: Duration): ComparableTimeMark
= LongTimeMark(startedAt, timeSource, offset + duration) \n override fun minus(other:
ComparableTimeMark): Duration { \n if (other !is LongTimeMark || this.timeSource != other.timeSource) \n
throw IllegalArgumentExpection("\nSubtracting or comparing time marks from different time sources is
not possible: $this and $other") \n\n // val thisValue = this.effectiveDuration() \n\n // val otherValue =
other.effectiveDuration() \n\n // if (thisValue == otherValue && thisValue.isInfinite()) return
Duration.ZERO \n\n // return thisValue - otherValue \n\n // if (this.offset == other.offset &&
this.offset.isInfinite()) return Duration.ZERO \n\n // val offsetDiff = this.offset - other.offset \n\n // val
startedAtDiff = (this.startedAt - other.startedAt).toDuration(timeSource.unit) \n\n // println("\n$startedAtDiff,
$offsetDiff") \n\n // return if (startedAtDiff == -offsetDiff) Duration.ZERO else startedAtDiff + offsetDiff \n
} \n\n override fun equals(other: Any?): Boolean = \n other is LongTimeMark && this.timeSource ==
other.timeSource && (this - other) == Duration.ZERO \n\n internal fun effectiveDuration(): Duration { \n
if (offset.isInfinite()) return offset \n val unit = timeSource.unit \n\n if (unit >= DurationUnit.MILLISECONDS) { \n
return startedAt.toDuration(unit) + offset \n } \n\n val scale = convertDurationUnit(1L, DurationUnit.MILLISECONDS, unit) \n\n val startedAtMillis =
startedAt / scale \n\n val startedAtRem = startedAt % scale \n\n return offset.toComponents {
offsetSeconds, offsetNanoseconds -> \n val offsetMillis = offsetNanoseconds / NANOS_IN_MILLIS \n\n val offsetRemNanos = offsetNanoseconds % NANOS_IN_MILLIS \n\n // add component-wise \n\n (startedAtRem.toDuration(unit) + offsetRemNanos.nanoseconds) + \n\n (startedAtMillis +
offsetMillis).milliseconds + \n\n offsetSeconds.seconds \n\n } \n\n } \n\n override fun

```

```

hashCode(): Int = effectiveDuration().hashCode()\n\n override fun toString(): String =
 \"LongTimeMark($startedAt${timeSource.unit.shortName()} + $offset
 (=${effectiveDuration()}), $timeSource)\"\\n\n }\\n\n\n override fun markNow(): ComparableTimeMark =
 LongTimeMark(read(), this, Duration.ZERO)\n\n}\n\n/**\n * An abstract class used to implement time sources that
 return their readings as [Double] values in the specified [unit].\n * @property unit The unit in which this time
 source's readings are expressed.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Using
 AbstractDoubleTimeSource is no longer recommended, use AbstractLongTimeSource instead.\")\npublic abstract
 class AbstractDoubleTimeSource(protected val unit: DurationUnit) : TimeSource.WithComparableMarks {\n
 /**\n * This protected method should be overridden to return the current reading of the time source expressed as a
 [Double] number\n * in the unit specified by the [unit] property.\n */\n protected abstract fun read():
 Double\n\n @Suppress(\"DEPRECATION\")\n private class DoubleTimeMark(private val startedAt: Double,
 private val timeSource:
 AbstractDoubleTimeSource, private val offset: Duration) : ComparableTimeMark {\n override fun
 elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n override fun
 plus(duration: Duration): ComparableTimeMark = DoubleTimeMark(startedAt, timeSource, offset + duration)\n\n override fun minus(other: ComparableTimeMark): Duration {\n if (other !is DoubleTimeMark ||
 this.timeSource != other.timeSource)\n throw IllegalArgumentException(\"Subtracting or comparing time
 marks from different time sources is not possible: $this and $other\")\n if (this.offset == other.offset &&
 this.offset.isInfinite()) return Duration.ZERO\n val offsetDiff = this.offset - other.offset\n val
 startedAtDiff = (this.startedAt - other.startedAt).toDuration(timeSource.unit)\n return if (startedAtDiff == -
 offsetDiff) Duration.ZERO else startedAtDiff + offsetDiff\n }\n\n override fun equals(other: Any?): Boolean {\n return other is DoubleTimeMark && this.timeSource ==
 other.timeSource && (this - other) == Duration.ZERO\n }\n\n override fun hashCode(): Int {\n
 return (startedAt.toDuration(timeSource.unit) + offset).hashCode()\n }\n\n override fun toString(): String =
 \"DoubleTimeMark($startedAt${timeSource.unit.shortName()} + $offset, $timeSource)\"\\n\n }\n\n override fun
 markNow(): ComparableTimeMark = DoubleTimeMark(read(), this, Duration.ZERO)\n\n}\n\n/**\n * A time source
 that has programmatically updatable readings. It is useful as a predictable source of time in tests.\n * @property
 current The current reading value can be advanced by the specified duration amount with the operator [plusAssign]:\n * val
 timeSource = TestTimeSource()\n * timeSource += 10.seconds\n * ```\n * Implementation note: the current
 reading value is stored as a [Long] number of nanoseconds,\n * thus it's capable to
 represent a time range of approximately \u00b1292 years.\n * Should the reading value overflow as the result of
 [plusAssign] operation, an [IllegalStateException] is thrown.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic class TestTimeSource : AbstractLongTimeSource(unit =
 DurationUnit.NANOSECONDS) {\n private var reading: Long = 0L\n\n override fun read(): Long = reading\n\n
 /**\n * Advances the current reading value of this time source by the specified [duration].\n * [duration]
 value is rounded down towards zero when converting it to a [Long] number of nanoseconds.\n * For example, if
 the duration being added is `0.6.nanoseconds`, the reading doesn't advance because\n * the duration value is
 rounded to zero nanoseconds.\n */\n @throws IllegalStateException when the reading value overflows as the
 result of this operation.\n public operator fun plusAssign(duration: Duration) {\n val longDelta =
 duration.toLong(unit)\n\n reading = if (longDelta != Long.MIN_VALUE && longDelta != Long.MAX_VALUE) {\n // when delta
 fits in long, add it as long\n val newReading = reading + longDelta\n if (reading xor longDelta >= 0
 && reading xor newReading < 0) overflow(duration)\n newReading\n } else {\n val delta =
 duration.toDouble(unit)\n // when delta is greater than long, add it as double\n val newReading =
 reading + delta\n if (newReading > Long.MAX_VALUE || newReading < Long.MIN_VALUE)\n overflow(duration)\n newReading.toLong()\n }\n }\n\n private fun overflow(duration: Duration) {\n
 throw IllegalStateException(\"TestTimeSource will overflow if its reading ${reading}${unit.shortName()} is
 advanced by $duration.\")\n }\n\n}\n\n\"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming

```

```

Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport
kotlin.time.Duration.Companion.milliseconds\nimport kotlin.time.Duration.Companion.nanoseconds\n\n// Long
time reading saturation math, shared between JVM and Native\n\ninternal fun saturatingAdd(longNs: Long,
duration: Duration): Long {\n val durationNs = duration.inWholeNanoseconds\n if (longNs.isSaturated()) { //
MIN_VALUE or MAX_VALUE - the reading is infinite\n return checkInfiniteSumDefined(longNs, duration,
durationNs)\n }\n if (durationNs.isSaturated()) { // duration doesn't fit in Long nanos\n return
saturatingAddInHalves(longNs, duration)\n }\n val result = longNs + durationNs\n if (((longNs xor result)
and (durationNs xor result)) < 0) {\n return if (longNs < 0) Long.MIN_VALUE else Long.MAX_VALUE\n }\n return result\n}\n\nprivate fun checkInfiniteSumDefined(longNs: Long, duration: Duration, durationNs:
Long): Long {\n if (duration.isInfinite()
&& (longNs xor durationNs < 0)) throw IllegalArgumentException("Summing infinities of different signs")\n return longNs\n}\n\nprivate fun saturatingAddInHalves(longNs: Long, duration: Duration): Long {\n val half =
duration / 2\n if (half.inWholeNanoseconds.isSaturated()) {\n // this will definitely saturate\n return
(longNs + duration.toDouble(DurationUnit.NANOSECONDS)).toLong()\n } else {\n return
saturatingAdd(saturatingAdd(longNs, half), duration - half)\n }\n}\n\ninternal fun saturatingDiff(valueNs: Long,
originNs: Long): Duration {\n if (originNs.isSaturated()) { // MIN_VALUE or MAX_VALUE\n return -
(originNs.toDuration(DurationUnit.DAYS)) // saturate to infinity\n }\n return saturatingFiniteDiff(valueNs,
originNs)\n}\n\ninternal fun saturatingOriginsDiff(origin1Ns: Long, origin2Ns: Long): Duration {\n if
(origin2Ns.isSaturated()) { // MIN_VALUE or MAX_VALUE\n if (origin1Ns == origin2Ns) return
Duration.ZERO //
saturated values of the same sign are considered equal\n return -(origin2Ns.toDuration(DurationUnit.DAYS)) //
saturate to infinity\n }\n if (origin1Ns.isSaturated()) {\n return origin1Ns.toDuration(DurationUnit.DAYS)\n }\n return saturatingFiniteDiff(origin1Ns, origin2Ns)\n}\n\nprivate fun saturatingFiniteDiff(value1Ns: Long,
value2Ns: Long): Duration {\n val result = value1Ns - value2Ns\n if ((result xor value1Ns) and (result xor
value2Ns).inv() < 0) {\n val resultMs = value1Ns / NANOS_IN_MILLIS - value2Ns / NANOS_IN_MILLIS\n val resultNs = value1Ns % NANOS_IN_MILLIS - value2Ns % NANOS_IN_MILLIS\n return
resultMs.milliseconds + resultNs.nanoseconds\n }\n return
result.nanoseconds\n}\n\n@Suppress("NOTHING_TO_INLINE")\nprivate inline fun Long.isSaturated(): Boolean
= (this - 1) or 1 == Long.MAX_VALUE // == either MAX_VALUE or MIN_VALUE\n", "/*\n * Copyright
2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
/\n\npackage kotlin.time\n\nimport kotlin.contracts.\n\n/**\n * Executes the given function [block] and returns the
duration of elapsed time interval.\n * The elapsed time is measured with [TimeSource.Monotonic].\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return
TimeSource.Monotonic.measureTime(block)\n}\n\n/**\n * Executes the given function [block] and returns the
duration of elapsed time interval.\n * The elapsed time is measured with the specified `this` [TimeSource]
instance.\n * @SinceKotlin("1.3")\n * @ExperimentalTime\n * public inline fun TimeSource.measureTime(block: ()
-> Unit): Duration {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n val
mark = markNow()\n block()\n return mark.elapsedNow()\n}\n\n/**\n * Executes the given function [block] and returns the duration
of elapsed time interval.\n * The elapsed time is measured with the specified `this` [TimeSource.Monotonic]
instance.\n * @SinceKotlin("1.7")\n * @ExperimentalTime\n * public inline fun
TimeSource.Monotonic.measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n val mark = markNow()\n block()\n return
mark.elapsedNow()\n}\n\n/**\n * Data class representing a result of executing an action, along with the duration
of elapsed time interval.\n * @property value the result of the action.\n * @property duration the time elapsed to

```

```

execute the action.\n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic data class TimedValue<T>(val
value: T, val duration: Duration)\n\n/**\n * Executes the given function [block] and returns an instance of
[TimedValue] class, containing both\n * the result
of the function execution and the duration of elapsed time interval.\n */\n * The elapsed time is measured with
[TimeSource.Monotonic].\n */\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T>
measureTimedValue(block: () -> T): TimedValue<T> {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n\n return
TimeSource.Monotonic.measureTimedValue(block)\n}\n\n/**\n * Executes the given [block] and returns an
instance of [TimedValue] class, containing both\n * the result of function execution and the duration of elapsed time
interval.\n */\n * The elapsed time is measured with the specified `this` [TimeSource] instance.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T> TimeSource.measureTimedValue(block: ()
-> T): TimedValue<T> {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n val mark = markNow()\n val result = block()\n return TimedValue(result, mark.elapsedNow())\n}\n\n/**\n * Executes the given [block] and returns an instance of [TimedValue] class, containing both\n * the result of
function execution and the duration of elapsed time interval.\n */\n * The elapsed time is measured with the specified
`this` [TimeSource.Monotonic] instance.\n */\n@SinceKotlin("1.7")\n@ExperimentalTime\npublic inline fun <T>
TimeSource.Monotonic.measureTimedValue(block: () -> T): TimedValue<T> {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n val mark = markNow()\n val result =
block()\n return TimedValue(result, mark.elapsedNow())\n}\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.native.concurrent.SharedImmutable\n\n/**\n * Defines deep recursive
function that
keeps its stack on the heap,\n * which allows very deep recursive computations that do not use the actual call
stack.\n * To initiate a call to this deep recursive function use its [invoke] function.\n * As a rule of thumb, it should
be used if recursion goes deeper than a thousand calls.\n */\n * The [DeepRecursiveFunction] takes one parameter of
type [T] and returns a result of type [R].\n * The [block] of code defines the body of a recursive function. In this
block\n * [callRecursive][DeepRecursiveScope.callRecursive] function can be used to make a recursive call\n * to
the declared function. Other instances of [DeepRecursiveFunction] can be called\n * in this scope with
`callRecursive` extension, too.\n */\n * For example, take a look at the following recursive tree class and a deeply\n *
recursive instance of this tree with 100K nodes:\n */\n * ```\n * class Tree(val left: Tree? = null, val right: Tree? =
null)\n * val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()\n * ```\n */\n * A regular recursive function can be defined to compute a depth of a tree:\n */\n * ```\n * fun depth(t:
Tree?): Int =\n * if (t == null) 0 else max(depth(t.left), depth(t.right)) + 1\n * println(depth(deepTree)) //
StackOverflowError\n * ```\n */\n * If this `depth` function is called for a `deepTree` it produces
`StackOverflowError` because of deep recursion.\n * However, the `depth` function can be rewritten using
`DeepRecursiveFunction` in the following way, and then\n * it successfully computes
[depth(deepTree)][DeepRecursiveFunction.invoke] expression:\n */\n * ```\n * val depth =
DeepRecursiveFunction<Tree?, Int> { t ->\n * if (t == null) 0 else max(callRecursive(t.left),
callRecursive(t.right)) + 1\n * }\n * println(depth(deepTree)) // Ok\n * ```\n */\n * Deep recursive functions can also
mutually call each other using a heap for the stack via\n * [callRecursive][DeepRecursiveScope.callRecursive]
extension. For example, the\n * following pair of mutually
recursive functions computes the number of tree nodes at even depth in the tree.\n */\n * ```\n * val mutualRecursion
= object {\n * val even: DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->\n * if (t ==
null) 0 else odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1\n * }\n * val odd:
DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->\n * if (t == null) 0 else
even.callRecursive(t.left) + even.callRecursive(t.right)\n * }\n * }\n * }\n * ```\n */\n * @param [T] the function
parameter type.\n * @param [R] the function result type.\n * @param block the function body.\n

```

```

*\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class
DeepRecursiveFunction<T, R>(\n internal val block: suspend DeepRecursiveScope<T, R>.(T) -> R)\n\n/**\n *
Initiates a call to this deep recursive function, forming a root of the call tree.\n * This operator should not be
used from inside of [DeepRecursiveScope] as
it uses the call stack slot for\n * initial recursive invocation. From inside of [DeepRecursiveScope] use\n *
[callRecursive][DeepRecursiveScope.callRecursive].\n
*\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic operator fun <T, R>
DeepRecursiveFunction<T, R>.invoke(value: T): R =\n DeepRecursiveScopeImpl<T, R>(block,
value).runCallLoop()\n\n/**\n * A scope class for [DeepRecursiveFunction] function declaration that defines
[callRecursive] methods to\n * recursively call this function or another [DeepRecursiveFunction] putting the call
activation frame on the heap.\n * @param [T] function parameter type.\n * @param [R] function result type.\n
*\n@RestrictsSuspension\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
sealed class DeepRecursiveScope<T, R> {\n /**\n * Makes recursive call to this [DeepRecursiveFunction]
function putting the call activation frame on the heap,\n * as opposed to the actual call
stack that is used by a regular recursive call.\n *\n * public abstract suspend fun callRecursive(value: T): R\n\n
/**\n * Makes call to the specified [DeepRecursiveFunction] function putting the call activation frame on the
heap,\n * as opposed to the actual call stack that is used by a regular call.\n *\n * public abstract suspend fun
<U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S\n\n @Deprecated(\n level =
DeprecationLevel.ERROR,\n message =\n "'invoke' should not be called from DeepRecursiveScope. '\n
+Use 'callRecursive' to do recursion in the heap instead of the call stack.'",\n replaceWith =
ReplaceWith("this.callRecursive(value)")\n)\n @Suppress("UNUSED_PARAMETER")\n public operator
fun DeepRecursiveFunction<*, *>.invoke(value: Any?): Nothing =\n throw
UnsupportedOperationException("Should not be called from DeepRecursiveScope")\n\n\n//
===== Implementation
=====
\nprivate typealias DeepRecursiveFunctionBlock = suspend DeepRecursiveScope<*,
*>.(Any?) -> Any?\n\n@SharedImmutable\nprivate val UNDEFINED_RESULT =
Result.success(COROUTINE_SUSPENDED)\n\n@Suppress("UNCHECKED_CAST")\nprivate class
DeepRecursiveScopeImpl<T, R>(\n block: suspend DeepRecursiveScope<T, R>.(T) -> R,\n value: T)\n :
DeepRecursiveScope<T, R>(), Continuation<R> {\n // Active function block\n private var function:
DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock\n // Value to call function with\n
private var value: Any? = value\n // Continuation of the current call\n private var cont: Continuation<Any?>?
= this as Continuation<Any?>\n // Completion result (completion of the whole call stack)\n private var result:
Result<Any?> = UNDEFINED_RESULT\n override val context: CoroutineContext\n get() =
EmptyCoroutineContext\n override fun resumeWith(result: Result<R>) {\n this.cont =
null\n this.result = result\n }\n override suspend fun callRecursive(value: T): R =
suspendCoroutineUninterceptedOrReturn { cont ->\n // calling the same function that is currently active\n
this.cont = cont as Continuation<Any?>\n this.value = value\n COROUTINE_SUSPENDED\n }\n override suspend fun <U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S =
suspendCoroutineUninterceptedOrReturn { cont ->\n // calling another recursive function\n val function =
block as DeepRecursiveFunctionBlock\n with(this@DeepRecursiveScopeImpl) {\n val currentFunction
= this.function\n if (function !== currentFunction) {\n // calling a different function -- create a
trampoline to restore function ref\n this.function = function\n this.cont =
crossFunctionCompletion(currentFunction, cont as Continuation<Any?>)\n } else {\n // calling the
same function -- direct\n this.cont = cont as Continuation<Any?>\n }\n this.value = value\n
 }\n COROUTINE_SUSPENDED\n }\n private fun crossFunctionCompletion(\n currentFunction:
DeepRecursiveFunctionBlock,\n cont: Continuation<Any?>\n): Continuation<Any?> =
Continuation(EmptyCoroutineContext) {\n this.function = currentFunction\n // When going back from a
trampoline we cannot just call cont.resume (stack usage!)\n // We delegate the cont.resumeWith(it) call to

```

```

runCallLoop\n this.cont = cont\n this.result = it\n }\n\n @Suppress(\"UNCHECKED_CAST\")\n fun
runCallLoop(): R {\n while (true) {\n // Note: cont is set to null in DeepRecursiveScopeImpl.resumeWith
when the whole computation completes\n val result = this.result\n val cont = this.cont\n ?:
return (result as Result<R>).getOrThrow() // done -- final result\n
 // The order of comparison is important here for that case of rogue class with broken equals\n if
(UNDEFINED_RESULT == result) {\n // call \"function\" with \"value\" using \"cont\" as completion\n
 val r = try {\n // This is block.startCoroutine(this, value, cont)\n
function.startCoroutineUninterceptedOrReturn(this, value, cont)\n } catch (e: Throwable) {\n
cont.resumeWithException(e)\n continue\n }\n // If the function returns without
suspension -- calls its continuation immediately\n if (r !== COROUTINE_SUSPENDED)\n
cont.resume(r as R)\n } else {\n // we returned from a crossFunctionCompletion trampoline -- call
resume here\n this.result = UNDEFINED_RESULT // reset result back\n
cont.resumeWith(result)\n }\n
 }\n }\n}\n\n\", \"/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n// Auto-generated file. DO NOT
EDIT!\n\n@file:kotlin.jvm.JvmName(\"NumbersKt\")\n@file:kotlin.jvm.JvmMultifileClass\npackage
kotlin\n\nimport kotlin.math.sign\n\n/** Divides this value by the other value, flooring the result to an integer that is
closer to negative infinity.
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic
inline fun Byte.mod(other: Byte): Byte = \n this.toInt().mod(other.toInt()).toByte()\n\n/**\n * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Short): Short = \n this.toInt().mod(other.toInt()).toShort()\n\n/**\n * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic
inline fun Byte.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Int): Int = \n this.toInt().mod(other)\n\n/**\n * Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n *
* The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute
value of the divisor.\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/**\n * Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.

```

```

*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other:
Byte): Byte = \n this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Short): Short = \n this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic
inline fun Short.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Int): Int = \n this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Byte): Int = \n this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.mod(other: Byte): Byte = \n this.mod(other.toInt()).toByte()\n\n/** Divides this value
by the other value, flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Short): Int = \n this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.mod(other: Short): Short = \n this.mod(other.toInt()).toShort()\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Int): Int {\n var q = this / other\n if (this xor other < 0 && q * other != this) q--\n return q\n}\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and

```

has the absolute value less than the absolute value of the divisor.

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Int): Int {\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 31))\n}\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity. */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Byte): Long = \n this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Byte): Byte = \n this.mod(other.toLong()).toByte()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Short): Long = \n this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Short): Short = \n this.mod(other.toLong()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Int): Long = \n this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Int): Int = \n this.mod(other.toLong()).toInt()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Long): Long {\n var q = this / other\n if (this xor other < 0 && q * other != this) q--\n return q\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Long): Long {\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 63))\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Float.mod(other: Float): Float {\n val r = this % other\n return if (r != 0.0.toFloat()) && r.sign != other.sign) r +
```



```

other else r\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The
result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of
the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In
this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Float.mod(other: Double): Double = \n this.toDouble().mod(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In
this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Double.mod(other: Float): Double = \n this.mod(other.toDouble())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Double.mod(other: Double): Double {\n val r = this % other\n return if (r != 0.0 && r.sign != other.sign) r +
other else r\n}\n\n"/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin\n\nimport kotlin.internal.InlineOnly\n\n/**\n * Returns a hash code value for the object or zero if the object
is `null`.\n * \n * @see Any.hashCode\n */\n@SinceKotlin("1.3")\n@InlineOnly\npublic inline fun
Any?.hashCode(): Int = this?.hashCode() ?: 0\n"/*\n
* Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Represents a version of the Kotlin standard
library.\n * \n * [major], [minor] and [patch] are integer components of a version,\n * they must be non-negative and
not greater than 255 ([MAX_COMPONENT_VALUE]).\n * \n * @constructor Creates a version from all three
components.\n
*\n@SinceKotlin("1.1")\npublic class KotlinVersion(val major: Int, val minor: Int, val patch: Int) :
Comparable<KotlinVersion> {\n /**\n * Creates a version from [major] and [minor] components, leaving
[patch] component zero.\n *\n * public constructor(major: Int, minor: Int) : this(major, minor, 0)\n *\n private val
version = versionOf(major, minor, patch)\n *\n private fun versionOf(major: Int, minor: Int, patch: Int): Int {\n
require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE && patch in
0..MAX_COMPONENT_VALUE) {\n \n"Version components are out of range: $major.$minor.$patch"\n
}\n return major.shl(16) + minor.shl(8) + patch\n }\n /**\n * Returns the string representation of this
version\n *\n * override fun toString(): String = "$major.$minor.$patch"\n *\n * override fun equals(other:
Any?): Boolean {\n if (this === other) return true\n val otherVersion = (other as?
KotlinVersion) ?: return false\n return this.version == otherVersion.version\n }\n *\n * override fun
hashCode(): Int = version\n *\n * override fun compareTo(other: KotlinVersion): Int = version - other.version\n
*\n * Returns `true` if this version is not less than the version specified\n * with the provided [major] and
[minor] components.\n *\n * public fun isAtLeast(major: Int, minor: Int): Boolean = // this.version >=
versionOf(major, minor, 0)\n this.major > major || (this.major == major &&\n this.minor >=
minor)\n *\n * Returns `true` if this version is not less than the version specified\n * with the provided
[major], [minor] and [patch] components.\n *\n * public fun isAtLeast(major: Int, minor: Int, patch: Int): Boolean
= // this.version >= versionOf(major, minor, patch)\n this.major > major || (this.major == major &&\n
(this.minor > minor || this.minor == minor &&\n

```

```

 this.patch >= patch))\n companion object {\n /**\n * Maximum value a version component can
have, a constant value 255.\n */\n // NOTE: Must be placed before CURRENT because its initialization
requires this field being initialized in JS\n public const val MAX_COMPONENT_VALUE = 255\n /**\n * Returns the current version of the Kotlin standard library.\n */\n @kotlin.jvm.JvmField\n public
val CURRENT: KotlinVersion = KotlinVersionCurrentValue.get()\n }\n\n// this class is ignored during
classpath normalization when considering whether to recompile dependencies in Kotlin build\nprivate object
KotlinVersionCurrentValue {\n @kotlin.jvm.JvmStatic\n fun get(): KotlinVersion = KotlinVersion(1, 8, 10) //
value is written here automatically during build\n}", "/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache
2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("LateinitKt")\n@file:Suppress("unused")\n\npackage kotlin\n\nimport
kotlin.internal.InlineOnly\nimport kotlin.internal.AccessibleLateinitPropertyLiteral\nimport
kotlin.reflect.KProperty0\n\n/**\n * Returns `true` if this lateinit property has been assigned a value, and `false`
otherwise.\n * Cannot be used in an inline function, to avoid binary compatibility issues.\n
*/\n\n@SinceKotlin("1.2")\n@InlineOnly\ninline val @receiver:AccessibleLateinitPropertyLiteral
KProperty0<*>.isInitialized: Boolean\n get() = throw NotImplementedError("Implementation is
intrinsic")\n}", "/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("LazyKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin\n\nimport
kotlin.reflect.KProperty\n\n/**\n * Represents a value with lazy initialization.\n * To create an instance of
[Lazy] use the [lazy] function.\n */\n\npublic interface Lazy<out T> {\n /**\n * Gets the lazily initialized value of
the current Lazy instance.\n * Once the value was initialized it must not change during the rest of lifetime of this
Lazy instance.\n */\n public val value: T\n /**\n * Returns `true` if a value for this Lazy instance has been
already initialized, and `false` otherwise.\n * Once this function has returned `true` it stays `true` for the rest of
lifetime of this Lazy instance.\n */\n public fun isInitialized(): Boolean\n}\n\n/**\n * Creates a new instance of
the [Lazy] that is already initialized with the specified [value].\n */\n\npublic fun <T> lazyOf(value: T): Lazy<T> =
InitializedLazyImpl(value)\n\n/**\n * An extension to delegate a read-only property of type [T] to an instance of
[Lazy].\n * This extension allows to
use instances of Lazy for property delegation:\n * `val property: String by lazy { initializer }`\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Lazy<T>.getValue(thisRef: Any?, property:
KProperty<*>): T = value\n\n/**\n * Specifies how a [Lazy] instance synchronizes initialization among multiple
threads.\n */\n\npublic enum class LazyThreadSafetyMode {\n /**\n * Locks are used to ensure that only a
single thread can initialize the [Lazy] instance.\n */\n SYNCHRONIZED,\n /**\n * Initializer function
can be called several times on concurrent access to uninitialized [Lazy] instance value,\n * but only the first
returned value will be used as the value of [Lazy] instance.\n */\n PUBLICATION,\n /**\n * No locks are
used to synchronize an access to the [Lazy] instance value; if the instance is accessed from multiple threads, its
behavior is undefined.\n * This mode should not be used unless the [Lazy] instance is guaranteed
never to be initialized from more than one thread.\n */\n NONE,\n}\n\n\ninternal object
UNINITIALIZED_VALUE\n\n// internal to be called from lazy in JS\ninternal class UnsafeLazyImpl<out
T>(initializer: () -> T) : Lazy<T>, Serializable {\n private var initializer: (() -> T)? = initializer\n private var
_value: Any? = UNINITIALIZED_VALUE\n\n override val value: T\n get() {\n if (_value ===
UNINITIALIZED_VALUE) {\n _value = initializer!!()\n initializer = null\n }\n
 }\n\n @Suppress("UNCHECKED_CAST")\n return _value as T\n }\n\n override fun isInitialized():
Boolean = _value !== UNINITIALIZED_VALUE\n\n override fun toString(): String = if (isInitialized())
value.toString() else "Lazy value not initialized yet.\n\n private fun writeReplace(): Any =
InitializedLazyImpl(value)\n}\n\ninternal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>,
Serializable {\n

```

```

 override fun isInitialized(): Boolean = true\n\n override fun toString(): String = value.toString()\n\n}\n\n"/*\n *
Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("NumbersKt")\npackage kotlin\n\n/**\n *
Counts the number of set bits in the binary representation of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary
representation of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation
of this [Int] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Int.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the
most significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeHighestOneBit(): Int\n\n/**\n * Returns a number having a single bit set in the position of the least
significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeLowestOneBit(): Int\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified
[bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least
significant bits on the right side.\n * Rotating the number left by a
negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more
generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.rotateLeft(bitCount: Int): Int\n\n/**\n * Rotates the binary representation of this [Int] number right by the
specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number
as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as
rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a
multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) ==
number.rotateRight(n % 32)`\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Int.rotateRight(bitCount: Int): Int\n\n/**\n * Counts the number of set bits in the binary representation
of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Long.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero
in the binary representation of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [Long] number.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [Long] number,\n * or zero, if this number
is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.takeHighestOneBit(): Long\n\n/**\n * Returns a number having a single bit set in the position of the least
significant set bit of this [Long] number,\n * or zero, if this number is zero.\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Long.takeLowestOneBit(): Long\n\n/**\n * Rotates the binary representation of this [Long] number left by the
specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as

```

the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [Long.SIZE\_BITS] (64) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateLeft(bitCount: Int): Long\n\n Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n
```

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateRight(bitCount: Int): Long\n\n Counts the number of set bits in the binary representation of this [Byte] number.\n
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n\n Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.\n
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)\n\n Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.\n
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits()\n\n Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number,\n * or zero, if this number is zero.\n
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()\n\n Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number,\n * or zero, if this number is zero.\n
```

```
\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()\n\n Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 8)`\n
```

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateLeft(bitCount: Int): Byte =\n (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()\n\n Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 8)`\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateRight(bitCount: Int): Byte =\n (toInt().shr(bitCount and 7) or (toInt() and 0xFF).shl(8 - (bitCount and 7))).toByte()\n
```

```

Int): Byte =\n (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n\n/**\n *
Counts the number of set bits in the binary representation of this [Short] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()\n\n/**\n * Counts the number of
consecutive most significant bits that are zero in the binary representation of this [Short] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countLeadingZeroBits(): Int =\n (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE_BITS
- Short.SIZE_BITS)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary
representation of this [Short] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c
inline fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits()\n\n/**\n * Returns a
number having a single bit set in the position of the most significant set bit of this [Short] number,\n * or zero, if this
number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n\n/**\n *
Returns a number having a single bit set in the position of the least significant set bit of this [Short] number,\n * or
zero, if this number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun Short.takeLowestOneBit(): Short = toInt().takeLowestOneBit().toShort()\n\n/**\n * Rotates the binary
representation of this [Short] number left by the specified [bitCount] number of bits.\n * The most significant bits
pushed out from
the left side reenter the number as the least significant bits on the right side.\n * \n * Rotating the number left by a
negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n * \n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more
generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
Short.rotateLeft(bitCount: Int): Short =\n (toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(16 - (bitCount
and 15))).toShort()\n\n/**\n * Rotates the binary representation of this [Short] number right by the specified
[bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the
most significant bits on the left side.\n * \n * Rotating the number right by a negative bit count is the same as rotating
it left by the negated bit count:\n
* `number.rotateRight(-n) == number.rotateLeft(n)`\n * \n * Rotating by a multiple of [Short.SIZE_BITS] (16)
returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
Short.rotateRight(bitCount: Int): Short =\n (toInt().shl(16 - (bitCount and 15)) or (toInt() and
0xFFFF).ushr(bitCount and 15)).toShort()\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\n\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.2")\n@Suppress("IN
VISIBLE_MEMBER", "INVISIBLE_REFERENCE")\npublic inline fun <R> suspend(noinline block: suspend ()
-> R): suspend () -> R = block\n\n"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmName("TuplesKt")\n\npackage kotlin\n\n/**\n * Represents a generic pair of two
values.\n * \n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Pair exhibits
value semantics, i.e. two pairs are equal if both components are equal.\n * \n * An example of decomposing it into
values:\n * @sample samples.misc.Tuples.pairDestructuring\n * \n * @param A type of the first value.\n * @param

```

```

B type of the second value.\n * @property first First value.\n * @property second Second value.\n * @constructor
Creates a new instance of Pair.\n */\npublic data class Pair<out A, out B>(\n public val first: A,\n public val\n second: B)\n) : Serializable {\n\n /**\n * Returns string representation of the [Pair] including its [first] and\n [second] values.\n\n */\n public override fun toString(): String = \"($first, $second)\"\n}\n\n/**\n * Creates a tuple of type [Pair] from\n this and [that].\n */\n * This can be useful for creating [Map] literals with less noise, for example:\n * @sample\n samples.collections.maps.instantiation.mapFromPairs\n */\npublic infix fun <A, B> A.to(that: B): Pair<A, B> =\n Pair(this, that)\n\n/**\n * Converts this pair into a list.\n */\n * @sample\n samples.misc.tuples.pairToList\n */\npublic\n fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)\n\n/**\n * Represents a triad of values\n */\n * There is\n no meaning attached to values in this class, it can be used for any purpose.\n * Triple exhibits value semantics, i.e.\n two triples are equal if all three components are equal.\n * An example of decomposing it into values:\n * @sample\n samples.misc.tuples.tripleDestructuring\n */\n * @param A type of the first value.\n * @param B type of the second\n value.\n * @param C type of the third value.\n * @property first First value.\n\n * @property second Second value.\n * @property third Third value.\n */\npublic data class Triple<out A, out B, out\n C>(\n public val first: A,\n public val second: B,\n public val third: C)\n) : Serializable {\n\n /**\n * Returns string representation of the [Triple] including its [first], [second] and [third] values.\n */\n public\n override fun toString(): String = \"($first, $second, $third)\"\n}\n\n/**\n * Converts this triple into a list.\n */\n * @sample\n samples.misc.tuples.tripleToList\n */\npublic\n fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first,\n second, third)\n\n", "/\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\npackage kotlin.ranges\n\nimport kotlin.internal.*\n\n/**\n * A\n range of values of type `UInt`.\n\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlib\n Api::class)\npublic\n class UIntRange(start: UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt>,\n OpenEndRange<UInt> {\n\n override val start: UInt get() = first\n override val endInclusive: UInt get() = last\n\n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @Deprecated("Can throw an exception when it's\n impossible to represent the value with UInt type, for example, when the range includes MAX_VALUE. It's\n recommended to use 'endInclusive' property that doesn't throw.")\n override val endExclusive: UInt get() {\n if (last == UInt.MAX_VALUE) error("Cannot return the exclusive upper bound of a range that includes\n MAX_VALUE.")\n return last + 1u\n }\n\n override fun contains(value: UInt): Boolean = first <= value\n && value <= last\n\n /**\n *\n Checks if the range is empty.\n */\n fun isEmpty(): Boolean = first > last\n\n /**\n *\n Checks if the range is empty if its start value is\n greater than\n the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other: Any?):\n Boolean =\n other is UIntRange && (isEmpty() && other.isEmpty() ||\n first == other.first && last ==\n other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * first.toInt() + last.toInt())\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n /**\n * An empty range of values of\n type UInt.\n */\n public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)\n }\n\n /**\n * A progression of values of type `UInt`.\n\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class\n UIntProgression\n internal constructor(\n start: UInt,\n endInclusive: UInt,\n step: Int)\n : Iterable<UInt> {\n\n init {\n if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step\n == Int.MIN_VALUE)\n throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on\n negation.")\n }\n\n /**\n * The first element in the progression.\n */\n public val first: UInt = start\n\n /**\n * The last element in the progression.\n */\n public val last: UInt = getProgressionLastElement(start,\n endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public val step: Int = step\n\n final\n override fun iterator(): Iterator<UInt> = UIntProgressionIterator(first, last, step)\n\n /**\n *\n Checks if the

```

```

progression is empty.\n \n * Progression with a positive step is empty if its first element is greater than the last
element.\n * Progression with a negative step is empty if its first element is less than the last element.\n */\n
public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun equals(other: Any?):
Boolean =\n other is
 UIntProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last && step
== other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt())
+ step.toInt())\n\n override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo
$last step ${-step}\""\n\n companion object {\n /**\n * Creates UIntProgression within the specified
bounds of a closed range.\n * The progression starts with the [rangeStart] value and goes toward the
[rangeEnd] value not excluding it, with the specified [step].\n * In order to go backwards the [step] must be
negative.\n *\n * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n */\n
 public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression =
 UIntProgression(rangeStart, rangeEnd, step)\n }\n}\n\n/**\n * An
iterator over a progression of values of type `UInt`.\n * @property step the number by which the value is
incremented on each step.\n */\n@SinceKotlin("1.3")\nprivate class UIntProgressionIterator(first: UInt, last: UInt,
step: Int) : Iterator<UInt> {\n private val finalElement = last\n private var hasNext: Boolean = if (step > 0) first
<= last else first >= last\n private val step = step.toInt() // use 2-complement math for negative steps\n private
var next = if (hasNext) first else finalElement\n override fun hasNext(): Boolean = hasNext\n\n override fun
next(): UInt {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw
kotlin.NoSuchElementException()\n hasNext = false\n } else {\n next += step\n }\n
return value\n }\n}\n\n"/**\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin.ranges\n\nimport kotlin.internal.*\n\n/**\n * A range of values of type `ULong`.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlib
Api::class)\npublic class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive,
1), ClosedRange<ULong>, OpenEndRange<ULong> {\n override val start: ULong get() = first\n override val
endInclusive: ULong get() = last\n\n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @Deprecated("Can throw an exception when it's impossible to represent the value with ULong type, for example,
when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.")\n override val endExclusive: ULong get() {\n if (last == ULong.MAX_VALUE) error("Cannot return the
exclusive upper bound of a range that includes
 MAX_VALUE.")\n return last + 1u\n }\n\n override fun contains(value: ULong): Boolean = first <= value
&& value <= last\n\n /**\n \n * Checks if the range is empty.\n \n * The range is empty if its start value is
greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other:
Any?): Boolean =\n other is ULongRange && (isEmpty() && other.isEmpty()) ||\n first == other.first
&& last == other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (first xor (first shr
32)).toInt() + (last xor (last shr 32)).toInt())\n\n override fun toString(): String = \"$first..$last"\n\n companion
object {\n /**\n * An empty range of values of type ULong.\n */\n public val EMPTY: ULongRange =
 ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)\n }\n}\n\n/**\n * A progression of values of type
`ULong`.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
open class ULongProgression\ninternal constructor(\n start: ULong,\n endInclusive: ULong,\n step: Long\n) :
Iterable<ULong> {\n init {\n if (step == 0.toLong()) throw kotlin.IllegalArgumentException("Step must be
non-zero.")\n if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater
than Long.MIN_VALUE to avoid overflow on negation.")\n }\n\n /**\n \n * The first element in the
progression.\n */\n public val first: ULong = start\n\n /**\n * The last element in the progression.\n */\n public val last: ULong = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the
progression.\n */\n public val step: Long = step\n\n final override fun iterator(): Iterator<ULong> =

```

```

ULongProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n\n *
Progression with a positive step is empty if its
first element is greater than the last element.\n * Progression with a negative step is empty if its first element is
less than the last element.\n\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first <
last\n\n override fun equals(other: Any?): Boolean =\n other is ULongProgression && (isEmpty() &&
other.isEmpty() ||\n first == other.first && last == other.last && step == other.step)\n\n override fun
hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * (first xor (first shr 32)).toInt() + (last xor (last shr
32)).toInt()) + (step xor (step ushr 32)).toInt())\n\n override fun toString(): String = if (step > 0) \"$first..$last step
$step\" else \"$first downTo $last step ${-step}\"
companion object {\n /**\n * Creates
ULongProgression within the specified bounds of a closed range.\n\n * The progression starts with the
[rangeStart] value and goes toward the [rangeEnd] value not excluding
it, with the specified [step].\n\n * In order to go backwards the [step] must be negative.\n\n * [step]
must be greater than `Long.MIN_VALUE` and not equal to zero.\n\n */\n public fun
fromClosedRange(rangeStart: ULong, rangeEnd: ULong, step: Long): ULongProgression =
ULongProgression(rangeStart, rangeEnd, step)\n }
}\n\n/**\n * An iterator over a progression of values of type
`ULong`.\n * @property step the number by which the value is incremented on each step.\n
*/\n@SinceKotlin("1.3")\nprivate class ULongProgressionIterator(first: ULong, last: ULong, step: Long) :
Iterator<ULong> {\n private val finalElement = last\n private var hasNext: Boolean = if (step > 0) first <= last
else first >= last\n private val step = step.toULong() // use 2-complement math for negative steps\n private var
next = if (hasNext) first else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun
next(): ULong {\n
val value = next\n if (value == finalElement) {\n if (!hasNext) throw
kotlin.NoSuchElementException()\n hasNext = false\n } else {\n next += step\n }\n
return value\n }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin.math\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: UInt, b: UInt): UInt {\n return minOf(a, b)\n}\n\n/**\n * Returns the smaller of two
values.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: ULong, b: ULong): ULong {\n return minOf(a, b)\n}\n\n/**\n * Returns the greater of
two values.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun max(a: UInt, b: UInt): UInt {\n return maxOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun max(a: ULong, b: ULong): ULong {\n return maxOf(a, b)\n}\n\n"/*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("UNumbersKt")\npackage kotlin\n/**\n * Counts the number of set bits in the
binary representation of this [UInt] number.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countOneBits(): Int =
toInt().countOneBits()\n\n/**\n * Counts
the number of consecutive most significant bits that are zero in the binary representation of this [UInt] number.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countLeadingZeroBits(): Int =
toInt().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the

```



binary representation of this [UInt] number.\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countTrailingZeroBits(): Int =
toInt().countTrailingZeroBits()\n\n/* Returns a number having a single bit set in the position of the most
significant set bit of this [UInt] number, or zero, if this number is zero.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()\n\n/* Returns a number
having a single bit set in the position of the least significant set bit of this [UInt] number, or zero, if this number
is zero.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeLowestOneBit(): UInt =
toInt().takeLowestOneBit().toUInt()\n\n/* Rotates the binary representation of this [UInt] number left by the
specified [bitCount] number of bits. The most significant bits pushed out from the left side reenter the number as
the least significant bits on the right side. Rotating the number left by a negative bit count is the same as
rotating it right by the negated bit count: number.rotateLeft(-n) == number.rotateRight(n)\n\n*\nRotating by a
multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally
number.rotateLeft(n)
== number.rotateLeft(n % 32)\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateLeft(bitCount: Int):
UInt = toInt().rotateLeft(bitCount).toUInt()\n\n/* Rotates the binary representation of this [UInt] number
right by the specified [bitCount] number of bits. The least significant bits pushed out from the right side reenter
the number as the most significant bits on the left side. Rotating the number right by a negative bit count is
the same as rotating it left by the negated bit count: number.rotateRight(-n) == number.rotateLeft(n)\n\n*\nRotating by a
multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally
number.rotateRight(n) == number.rotateRight(n % 32)\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline
fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()\n\n/* Counts the number of
set bits in the binary representation of this [ULong] number.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countOneBits(): Int =
toLong().countOneBits()\n\n/* Counts the number of consecutive most significant bits that are zero in the
binary representation of this [ULong] number.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countLeadingZeroBits(): Int =
toLong().countLeadingZeroBits()\n\n/* Counts the number of consecutive least significant bits that are zero
in the binary representation of this [ULong] number.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()\n\n/* Returns a number
having a single bit set in the position of the most significant set bit of this [ULong] number, or zero, if this
number is zero.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeHighestOneBit(): ULong
= toLong().takeHighestOneBit().toULong()\n\n/* Returns a number having a single bit set in the position of the
least significant set bit of this [ULong] number, or zero, if this number is zero.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeLowestOneBit(): ULong
= toLong().takeLowestOneBit().toULong()\n\n/* Rotates the binary representation of this [ULong] number left
by the specified [bitCount]
```



```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateRight(bitCount:
Int): UByte = toByte().rotateRight(bitCount).toUByte()\n\n/**\n * Counts the number of set bits in the
binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countOneBits(): Int =
toUInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countLeadingZeroBits(): Int =
toShort().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countTrailingZeroBits(): Int =
toShort().countTrailingZeroBits()\n\n/**\n * Returns
a number having a single bit set in the position of the most significant set bit of this [UShort] number,\n * or zero, if
this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeHighestOneBit(): UShort =
toInt().takeHighestOneBit().toUShort()\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeLowestOneBit(): UShort =
toInt().takeLowestOneBit().toUShort()\n\n/**\n * Rotates the binary representation of this [UShort] number left
by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant
bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the
negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of
[UShort.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) ==
number.rotateLeft(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateLeft(bitCount:
Int): UShort = toShort().rotateLeft(bitCount).toUShort()\n\n/**\n * Rotates the binary representation of this
[UShort] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the
right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a
negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) ==
number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateRight(bitCount:
Int): UShort = toShort().rotateRight(bitCount).toUShort()\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n@npackage kotlin.internal\n@n// (a - b) mod c\nprivate fun
differenceModulo(a: UInt, b: UInt, c: UInt): UInt {\n val ac = a % c\n val bc = b % c\n return if (ac >= bc) ac -
bc else ac - bc + c\n}\n\nprivate fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {\n val ac = a %
c\n val bc = b % c\n return if (ac >= bc) ac - bc else ac - bc + c\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is
in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n *
[step].\n * No validation on passed parameters is performed. The given parameters should satisfy the

```

```

condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start
first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or
difference of successive elements in the progression\n * @return the final element of the progression\n *
@suppress\n */\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: UInt, end:
UInt, step: Int): UInt = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start,
step.toUInt())\n step < 0 -> if (start <= end) end else end + differenceModulo(start,
end, (-step).toUInt())\n else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n\n/**\n * Calculates
the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n
* from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No
validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either
`step > 0` and `start <= end`,\n *\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the
progression\n * @param end ending bound for the progression\n * @param step increment, or difference of
successive elements in the progression\n * @return the final element of the progression\n * @suppress\n
*/\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: ULong, end: ULong,
step: Long): ULong = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end,
start, step.toULong())\n step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-
step).toULong())\n else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n\n"/**\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("UStringsKt") // string representation of unsigned numbers\n\npackage
kotlin.text\n\n/**\n * Returns a string representation of this [Byte] value in the specified [radix].\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string
representation of this [Short] value in the specified
[radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string
conversion.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string
representation of this [Int] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is
not a valid radix for number to string conversion.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)\n\n/**\n * Returns a string
representation of this [Long] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix]
is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))\n\n/**\n * Parses the
string as a signed [UByte] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a
valid representation of a number.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte():
UByte = toUByteOrNull() ?: numberFormatException(this)\n\n/**\n * Parses the string as a signed [UByte] number and
returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*/\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatException(this)\n\n/**\n * Parses the
string as a [UShort] number and returns the result.\n *\n * @throws NumberFormatException if the
string is not a valid representation of a number.\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort():

```



```

limitForMaxRadix = 119304647u // limit / 36\n\n var limitBeforeMul = limitForMaxRadix\n val uradix =
radix.toUInt()\n var result = 0u\n for (i
in start until length) {\n val digit = digitOf(this[i], radix)\n\n if (digit < 0) return null\n\n if (result >
limitBeforeMul) {\n\n if (limitBeforeMul == limitForMaxRadix) {\n\n limitBeforeMul = limit /
uradix\n\n\n if (result > limitBeforeMul) {\n\n return null\n\n\n }\n\n } else {\n
return null\n\n\n }\n\n\n result *= uradix\n\n val beforeAdding = result\n\n result +=
digit.toUInt()\n\n if (result < beforeAdding) return null // overflow has happened\n\n }\n\n return
result\n\n}\n\n**\n * Parses the string as an [ULong] number and returns the result\n * or `null` if the string is not a
valid representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)\n\n**\n * Parses the string as an [ULong] number
and returns the result\n
* or `null` if the string is not a valid representation of a number.\n * \n * @throws IllegalArgumentException when
[radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(radix: Int): ULong? {\n checkRadix(radix)\n\n val length = this.length\n\n if (length ==
0) return null\n\n val limit: ULong = ULong.MAX_VALUE\n\n val start: Int\n\n val firstChar = this[0]\n\n if
(firstChar < '0') {\n\n if (length == 1 || firstChar != '+') return null\n\n start = 1\n\n } else {\n\n start = 0\n
}\n\n\n val limitForMaxRadix = 512409557603043100uL // limit / 36\n\n var limitBeforeMul =
limitForMaxRadix\n\n val uradix = radix.toULong()\n\n var result = 0uL\n\n for (i in start until length) {\n\n val
digit = digitOf(this[i], radix)\n\n\n if (digit < 0) return null\n\n\n if (result > limitBeforeMul) {\n\n if
(limitBeforeMul
== limitForMaxRadix) {\n\n\n limitBeforeMul = limit / uradix\n\n\n\n if (result > limitBeforeMul) {\n\n
return null\n\n\n\n }\n\n\n } else {\n\n\n return null\n\n\n\n }\n\n\n }\n\n\n result *=
uradix\n\n\n val beforeAdding = result\n\n\n result += digit.toUInt()\n\n\n if (result < beforeAdding) return null
// overflow has happened\n\n\n }\n\n\n return result\n\n}\n\n**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n * \n@file:Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\nimport
kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\n\n**\n * Marks the API that is
dependent on the experimental unsigned types, including those types themselves.\n
*\n * Usages of such API will be reported as warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g.
`@OptIn(ExperimentalUnsignedTypes::class)`,\n * or with the `-opt-in=kotlin.ExperimentalUnsignedTypes`
compiler option is given.\n * \n * It's recommended to propagate the experimental status to the API that depends on
unsigned types by annotating it with this annotation.\n * \n@RequiresOptIn(level =
RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Target(CLASS, ANNOTATION_CLASS,
PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION,
PROPERTY_GETTER, PROPERTY_SETTER,
TYPEALIAS)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class
ExperimentalUnsignedTypes\n\n**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MathKt")\n\npackage
kotlin.math\n\n\n// constants, can't use them from nativeMath as they are not constants there\n\n**\n * Ratio of the
circumference of a circle to its diameter, approximately 3.14159. *\n@SinceKotlin("1.2")\n\npublic const val PI:
Double = 3.141592653589793\n\n**\n * Base of the natural logarithms, approximately 2.71828.
*\n@SinceKotlin("1.2")\n\npublic const val E: Double = 2.718281828459045\n\n// region =====
Double Math =====\n\n**\n * Computes the sine of the angle [x]

```

given in radians.  
`sin`: `Double` \* Special cases: `sin(NaN|+Inf|-Inf)` is `NaN`  
`cos`: `Double` \* Special cases: `cos(NaN|+Inf|-Inf)` is `NaN`  
`tan`: `Double` \* Special cases: `tan(NaN|+Inf|-Inf)` is `NaN`  
`asin`: `Double` \* Special cases: `asin(x)` is `NaN`, when `abs(x) > 1` or `x` is `NaN`  
`acos`: `Double` \* Special cases: `acos(x)` is `NaN`, when `abs(x) > 1` or `x` is `NaN`  
`atan`: `Double` \* Special cases: `atan(NaN)` is `NaN`  
`atan2`: `Double, Double` \* Special cases: `atan2(0.0, 0.0)` is `0.0`, `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`, `atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`, `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for `-Inf < y < 0`, `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`, `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`, `atan2(+Inf, x)` is `PI/2` for finite `x`, `atan2(-Inf, x)` is `-PI/2` for finite `x`, `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`  
`sinh`: `Double` \* Special cases: `sinh(NaN)` is `NaN`, `sinh(+Inf)` is `+Inf`, `sinh(-Inf)` is `-Inf`  
`cosh`: `Double` \* Special cases: `cosh(NaN)` is `NaN`, `cosh(+Inf|-Inf)` is `+Inf`  
`tanh`: `Double` \* Special cases: `tanh(NaN)` is `NaN`, `tanh(+Inf)` is `1.0`, `tanh(-Inf)` is `-1.0`  
`asinh`: `Double` \* Special cases: `asinh(NaN)` is `NaN`, `asinh(+Inf)` is `+Inf`, `asinh(-Inf)` is `-Inf`  
`acosh`: `Double` \* Special cases: `acosh(NaN)` is `NaN`, `acosh(x)` is `NaN` when `x < 1`, `acosh(+Inf)` is `+Inf`  
`atanh`: `Double` \* Special cases: `atanh(NaN)` is `NaN`, `atanh(x)` is `NaN` when `x > 1` or `x < -1`, `atanh(1.0)` is `+Inf`, `atanh(-1.0)` is `-Inf`  
`hypot`: `Double, Double` \* Special cases: `hypot(x, y)` is `NaN` when `x < 0` or `y < 0`  
`exp`: `Double` \* Special cases: `exp(NaN)` is `NaN`, `exp(+Inf)` is `+Inf`, `exp(-Inf)` is `0.0`  
`expm1`: `Double` \* Special cases: `expm1(NaN)` is `NaN`, `expm1(+Inf)` is `+Inf`, `expm1(-Inf)` is `-1.0`  
`log`: `Double, Double` \* Special cases: `log(x, base)` is `NaN` when `x < 0` or `base < 0` or `base == 1`

-  $\log(x, b)$  is NaN if either  $x$  or  $b$  are NaN \*  $-\log(x, b)$  is NaN when  $x < 0$  or  $b \leq 0$  or  $b == 1.0$  \*  $-\log(+Inf, +Inf)$  is NaN \*  $-\log(+Inf, b)$  is +Inf for  $b > 1$  and  $-\text{Inf}$  for  $b < 1$  \*  $-\log(0.0, b)$  is  $-\text{Inf}$  for  $b > 1$  and  $+\text{Inf}$  for  $b > 1$  \* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

```

public expect fun log(x: Double, base: Double): Double
 Computes the natural logarithm (base 'E') of the value [x].
 Special cases:
 - ln(NaN) is NaN
 - ln(x) is NaN when x < 0.0
 - ln(+Inf) is +Inf
 - ln(0.0) is -Inf

```

```

public expect fun ln(x: Double): Double
 Computes the common logarithm (base 10) of the value [x].
 @see [ln] function for special cases.

```

```

public expect fun log10(x: Double): Double
 Computes the binary logarithm (base 2) of the value [x].
 @see [ln] function for special cases.

```

```

public expect fun log2(x: Double): Double
 Computes ln(x + 1).
 This function can be implemented to produce more precise result for [x] near zero.
 Special cases:
 - ln1p(NaN) is NaN
 - ln1p(x) is NaN where x < -1.0
 - ln1p(-1.0) is -Inf
 - ln1p(+Inf) is +Inf
 @see [ln] function
 @see [expm1] function

```

```

public expect fun ln1p(x: Double): Double
 Rounds the given value [x] to an integer towards positive infinity.
 @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.
 Special cases:
 - ceil(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

public expect fun ceil(x: Double): Double
 Rounds the given value [x] to an integer towards negative infinity.
 @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.
 Special cases:
 - floor(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

public expect fun floor(x: Double): Double
 Rounds the given value [x] to an integer towards zero.
 @return the value [x] having its fractional part truncated.
 Special cases:
 - truncate(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

public expect fun truncate(x: Double): Double
 Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
 Special cases:
 - round(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

public expect fun round(x: Double): Double
 Returns the absolute value of the given value [x].
 Special cases:
 - abs(NaN) is NaN
 @see absoluteValue extension property for [Double]

```

```

public expect fun abs(x: Double): Double
 Returns the sign of the given value [x]:
 - -1.0 if the value is negative,
 - zero if the value is zero,
 - 1.0 if the value is positive
 Special case:
 - sign(NaN) is NaN

```

```

public expect fun sign(x: Double): Double
 Returns the smaller of two values.
 If either value is NaN, then the result is NaN.

```

```

public expect fun min(a: Double, b: Double): Double
 Returns the greater of two values.
 If either value is NaN, then the result is NaN.

```

```

public expect fun max(a: Double, b: Double): Double
 Returns the cube root of [x]. For any x, cbrt(-x) == -cbrt(x); that is, the cube root of a negative value is the negative of the cube root of that value's magnitude.
 Special cases:
 - If the argument is NaN, then the result is NaN.
 - If the argument is infinite, then the result is an infinity with the same sign as the argument.
 - If the argument is zero, then the result is a zero with the same sign as the argument.

```

```

@WasExperimental(ExperimentalStdlibApi::class)
public expect fun cbrt(x: Double): Double
 Raises this value to the power [x].
 Special cases:
 - b.pow(0.0) is 1.0
 - b.pow(1.0) == b
 - b.pow(NaN) is NaN
 - NaN.pow(x) is NaN for x != 0.0
 - b.pow(Inf) is NaN for abs(b) == 1.0
 - b.pow(x) is NaN for b < 0 and x is finite and not an integer

```

```

public expect fun Double.pow(x: Double): Double
 Raises this value to the integer power [n].
 See the other overload of [pow] for details.

```

```

public expect fun Double.pow(n: Int): Double
 Returns the absolute value of this value.
 Special cases:
 - NaN.absoluteValue is NaN
 @see abs

```



function\n \* \n@SinceKotlin("1.2")\npublic expect val Double.absoluteValue: Double\n\n/\*\*\n \* Returns the sign of this value:\n \* -  $-1.0$  if the value is negative,\n \* - zero if the value is zero,\n \* -  $1.0$  if the value is positive\n \* \n \* Special case:\n \* -  $\text{NaN.sign}$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect val Double.sign: Double\n\n/\*\*\n \* Returns this value with the sign bit same as of the [sign] value.\n \* \n \* If [sign] is  $\text{NaN}$  the sign of the result is undefined.\n \* \n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Double): Double\n\n/\*\*\n \* Returns this value with the sign bit same as of the [sign] value.\n \* \n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Int): Double\n\n/\*\*\n \* Returns the ulp (unit in the last place) of this value.\n \* \n \* An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.\n \* \n \* Special Cases:\n \* -  $\text{NaN.ulp}$  is  $\text{NaN}$ \n \* -  $x.ulp$  is  $+\text{Inf}$  when  $x$  is  $+\text{Inf}$  or  $-\text{Inf}$ \n \* -  $0.0.ulp$  is  $\text{Double.MIN\_VALUE}$ \n \* \n@SinceKotlin("1.2")\npublic expect val Double.ulp: Double\n\n/\*\*\n \* Returns the [Double] value nearest to this value in direction of positive infinity.\n \* \n@SinceKotlin("1.2")\npublic expect fun Double.nextUp(): Double\n\n/\*\*\n \* Returns the [Double] value nearest to this value in direction of negative infinity.\n \* \n@SinceKotlin("1.2")\npublic expect fun Double.nextDown(): Double\n\n/\*\*\n \* Returns the [Double] value nearest to this value in direction from this value towards the value [to].\n \* \n \* Special cases:\n \* -  $x.nextTowards(y)$  is  $\text{NaN}$  if either  $x$  or  $y$  are  $\text{NaN}$ \n \* -  $x.nextTowards(x) == x$ \n \* \n@SinceKotlin("1.2")\npublic expect fun Double.nextTowards(to: Double): Double\n\n/\*\*\n \* Rounds this [Double] value to the nearest integer and converts the result to [Int].\n \* \n \* Ties are rounded towards positive infinity.\n \* \n \* Special cases:\n \* -  $x.roundToInt() == \text{Int.MAX\_VALUE}$  when  $x > \text{Int.MAX\_VALUE}$ \n \* -  $x.roundToInt() == \text{Int.MIN\_VALUE}$  when  $x < \text{Int.MIN\_VALUE}$ \n \* \n \* @throws IllegalArgumentException when this value is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun Double.roundToInt(): Int\n\n/\*\*\n \* Rounds this [Double] value to the nearest integer and converts the result to [Long].\n \* \n \* Ties are rounded towards positive infinity.\n \* \n \* Special cases:\n \* -  $x.roundToLong() == \text{Long.MAX\_VALUE}$  when  $x > \text{Long.MAX\_VALUE}$ \n \* -  $x.roundToLong() == \text{Long.MIN\_VALUE}$  when  $x < \text{Long.MIN\_VALUE}$ \n \* \n \* @throws IllegalArgumentException when this value is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun Double.roundToLong(): Long\n\n// endregion\n\n// region\n\n==== Float Math =====\n\n/\*\*\n \* Computes the sine of the angle [x] given in radians.\n \* \n \* Special cases:\n \* -  $\text{sin}(\text{NaN}|\text{+Inf}|\text{-Inf})$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun sin(x: Float): Float\n\n/\*\*\n \* Computes the cosine of the angle [x] given in radians.\n \* \n \* Special cases:\n \* -  $\text{cos}(\text{NaN}|\text{+Inf}|\text{-Inf})$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun cos(x: Float): Float\n\n/\*\*\n \* Computes the tangent of the angle [x] given in radians.\n \* \n \* Special cases:\n \* -  $\text{tan}(\text{NaN}|\text{+Inf}|\text{-Inf})$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun tan(x: Float): Float\n\n/\*\*\n \* Computes the arc sine of the value [x];\n \* \n \* the returned value is an angle in the range from  $-\text{PI}/2$  to  $\text{PI}/2$  radians.\n \* \n \* Special cases:\n \* -  $\text{asin}(x)$  is  $\text{NaN}$ , when  $\text{abs}(x) > 1$  or  $x$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun asin(x: Float): Float\n\n/\*\*\n \* Computes the arc cosine of the value [x];\n \* \n \* the returned value is an angle in the range from  $0.0$  to  $\text{PI}$  radians.\n \* \n \* Special cases:\n \* -  $\text{acos}(x)$  is  $\text{NaN}$ , when  $\text{abs}(x) > 1$  or  $x$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun acos(x: Float): Float\n\n/\*\*\n \* Computes the arc tangent of the value [x];\n \* \n \* the returned value is an angle in the range from  $-\text{PI}/2$  to  $\text{PI}/2$  radians.\n \* \n \* Special cases:\n \* -  $\text{atan}(\text{NaN})$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun atan(x: Float): Float\n\n/\*\*\n \* Returns the angle  $\theta$  of the polar coordinates  $(r, \theta)$  that correspond\n \* to the rectangular coordinates  $(x, y)$  by computing the arc tangent of the value  $[y] / [x]$ ;\n \* \n \* the returned value is an angle in the range from  $-\text{PI}$  to  $\text{PI}$  radians.\n \* \n \* Special cases:\n \* -  $\text{atan2}(0.0, 0.0)$  is  $0.0$ \n \* -  $\text{atan2}(0.0, x)$  is  $0.0$  for  $x > 0$  and  $\text{PI}$  for  $x < 0$ \n \* -  $\text{atan2}(-0.0, x)$  is  $-0.0$  for  $x > 0$  and  $-\text{PI}$  for  $x < 0$ \n \* -  $\text{atan2}(y, +\text{Inf})$  is  $0.0$  for  $0 < y < +\text{Inf}$  and  $-0.0$  for  $-\text{Inf} < y < 0$ \n \* -  $\text{atan2}(y, -\text{Inf})$  is  $\text{PI}$  for  $0 < y < +\text{Inf}$  and  $-\text{PI}$  for  $-\text{Inf} < y < 0$ \n \* -  $\text{atan2}(y, 0.0)$  is  $\text{PI}/2$  for  $y > 0$  and  $-\text{PI}/2$  for  $y < 0$ \n \* -  $\text{atan2}(+\text{Inf}, x)$  is  $\text{PI}/2$  for finite  $x \neq 0$ \n \* -  $\text{atan2}(-\text{Inf}, x)$  is  $-\text{PI}/2$  for finite  $x \neq 0$ \n \* -  $\text{atan2}(\text{NaN}, x)$  and  $\text{atan2}(y, \text{NaN})$  is  $\text{NaN}$ \n \* \n@SinceKotlin("1.2")\npublic expect fun atan2(y: Float, x: Float): Float\n\n/\*\*\n \* Computes the hyperbolic sine of the value [x].\n \* \n \* Special cases:\n \* -  $\text{sinh}(\text{NaN})$  is  $\text{NaN}$ \n \* -  $\text{sinh}(+\text{Inf})$

`is`+Inf`n * -`sinh(-Inf)` is`-Inf`n */n@SinceKotlin("1.2")\npublic expect fun sinh(x: Float): Float\n/n/**\n * Computes the hyperbolic cosine of the value [x].\n *n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * -  
`cosh(+Inf|-Inf)` is`+Inf`n */n@SinceKotlin("1.2")\npublic expect fun cosh(x: Float): Float\n/n/**\n * Computes the hyperbolic tangent of the value [x].\n *n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is  
`1.0`\n * - `tanh(-Inf)` is`-1.0`\n */n@SinceKotlin("1.2")\npublic expect fun tanh(x: Float): Float\n/n/**\n * Computes the inverse hyperbolic sine of the value [x].\n *n * The returned value is `y` such that `sinh(y) == x`.\n *n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)`  
is`+Inf`n * - `asinh(-Inf)` is`-Inf`n */n@SinceKotlin("1.2")\npublic expect fun asinh(x: Float): Float\n/n/**\n * Computes the inverse hyperbolic cosine of the value [x].\n *n * The returned value is positive `y` such that  
`cosh(y) == x`.\n *n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * -  
`acosh(+Inf)` is`+Inf`n */n@SinceKotlin("1.2")\npublic expect fun acosh(x: Float): Float\n/n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n *n * The returned value is `y` such that `tanh(y) == x`.\n *n *  
Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is  
`+Inf`\n * - `tanh(-1.0)` is`-Inf`n */n@SinceKotlin("1.2")\npublic expect fun atanh(x: Float): Float\n/n/**\n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n *n * Special cases:\n * - returns `+Inf`  
if any of arguments is infinite\n * - returns `NaN`  
if any of arguments is `NaN` and the other is not infinite\n */n@SinceKotlin("1.2")\npublic expect fun hypot(x:  
Float, y: Float): Float\n/n/**\n * Computes the positive square root of the value [x].\n *n * Special cases:\n * -  
`sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`\n */n@SinceKotlin("1.2")\npublic expect fun sqrt(x: Float):  
Float\n/n/**\n * Computes Euler's number `e` raised to the power of the value [x].\n *n * Special cases:\n * -  
`exp(NaN)` is `NaN`\n * - `exp(+Inf)` is`+Inf`\n * - `exp(-Inf)` is`0.0`\n */n@SinceKotlin("1.2")\npublic  
expect fun exp(x: Float): Float\n/n/**\n * Computes `exp(x) - 1`.\n *n * This function can be implemented to  
produce more precise result for [x] near zero.\n *n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * -  
`expm1(+Inf)` is`+Inf`\n * - `expm1(-Inf)` is`-1.0`\n *n * @see [exp] function.\n  
*/n@SinceKotlin("1.2")\npublic expect fun expm1(x: Float): Float\n/n/**\n * Computes the logarithm of the  
value [x] to  
the given [base].\n *n * Special cases:\n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n * - `log(x, b)` is  
`NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n * - `log(+Inf, +Inf)` is `NaN`\n * - `log(+Inf, b)` is`+Inf` for `b >  
1` and`-Inf` for `b < 1`\n * - `log(0.0, b)` is`-Inf` for `b > 1` and`+Inf` for `b > 1`\n *n * See also logarithm  
functions for common fixed bases: [ln], [log10] and [log2].\n */n@SinceKotlin("1.2")\npublic expect fun log(x:  
Float, base: Float): Float\n/n/**\n * Computes the natural logarithm (base `E`) of the value [x].\n *n * Special  
cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(+Inf)` is`+Inf`\n * - `ln(0.0)` is`-  
Inf`n */n@SinceKotlin("1.2")\npublic expect fun ln(x: Float): Float\n/n/**\n * Computes the common logarithm  
(base 10) of the value [x].\n *n * @see [ln] function for special cases.\n */n@SinceKotlin("1.2")\npublic expect  
fun log10(x: Float): Float\n/n/**\n * Computes the  
binary logarithm (base 2) of the value [x].\n *n * @see [ln] function for special cases.\n  
*/n@SinceKotlin("1.2")\npublic expect fun log2(x: Float): Float\n/n/**\n * Computes `ln(x + 1)`.\n *n * This  
function can be implemented to produce more precise result for [x] near zero.\n *n * Special cases:\n * -  
`ln1p(NaN)` is `NaN`\n * - `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is`-Inf`\n * - `ln1p(+Inf)` is  
`+Inf`\n *n * @see [ln] function\n * @see [expm1] function\n */n@SinceKotlin("1.2")\npublic expect fun  
ln1p(x: Float): Float\n/n/**\n * Rounds the given value [x] to an integer towards positive infinity.\n *n * @return the  
smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.\n *n * Special  
cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or`-Inf` or already a mathematical integer.\n  
*/n@SinceKotlin("1.2")\npublic expect fun ceil(x: Float): Float\n/n/**\n * Rounds the given value [x] to an  
integer towards negative infinity.\n *n * @return the largest Float value that is smaller than or equal to the given  
value [x] and is a mathematical integer.\n *n * Special cases:\n * - `floor(x)` is `x` where `x` is `NaN` or`+Inf` or  
`-Inf` or already a mathematical integer.\n */n@SinceKotlin("1.2")\npublic expect fun floor(x: Float):  
Float\n/n/**\n * Rounds the given value [x] to an integer towards zero.\n *n * @return the value [x] having its`

fractional part truncated.  
`* Special cases:`  
`- truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
`* Since Kotlin("1.2")`  
`public expect fun truncate(x: Float): Float`  
Rounds the given value [x] towards the closest integer with ties rounded towards even integer.  
`* Special cases:`  
`- round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.  
`* Since Kotlin("1.2")`  
`public expect fun round(x: Float): Float`  
Returns the absolute value of the given value [x].  
`* Special cases:`  
`- abs(NaN)` is `NaN`  
`* @see absoluteValue` extension property for [Float]  
`* Since Kotlin("1.2")`  
`public expect fun abs(x: Float): Float`  
Returns the sign of the given value [x]:  
`- -1.0` if the value is negative,  
`- zero` if the value is zero,  
`- 1.0` if the value is positive  
`* Special case:`  
`- sign(NaN)` is `NaN`  
`* Since Kotlin("1.2")`  
`public expect fun sign(x: Float): Float`  
Returns the smaller of two values.  
`* If either value is NaN, then the result is NaN.`  
`* Since Kotlin("1.2")`  
`public expect fun min(a: Float, b: Float): Float`  
Returns the greater of two values.  
`* If either value is NaN, then the result is NaN.`  
`* Since Kotlin("1.2")`  
`public expect fun max(a: Float, b: Float): Float`  
Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`; that is, the cube root of a negative value is the negative of the cube root of that value's magnitude. Special cases:  
`* Special cases:`  
`- If the argument is NaN, then the result is NaN.`  
`- If the argument is infinite, then the result is an infinity with the same sign as the argument.`  
`- If the argument is zero, then the result is a zero with the same sign as the argument.`  
`* Since Kotlin("1.8")`  
`@WasExperimental(ExperimentalStdlibApi::class)`  
`public expect fun cbrt(x: Float): Float`  
// extensions  
`* Raises this value to the power [x].`  
`* Special cases:`  
`- b.pow(0.0)` is `1.0`  
`- b.pow(1.0) == b`  
`- b.pow(NaN)` is `NaN`  
`- NaN.pow(x)` is `NaN` for `x != 0.0`  
`- b.pow(Inf)` is `NaN` for `abs(b) == 1.0`  
`- b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer  
`* Since Kotlin("1.2")`  
`public expect fun Float.pow(x: Float): Float`  
Raises this value to the integer power [n].  
`* See the other overload of [pow] for details.`  
`* Since Kotlin("1.2")`  
`public expect fun Float.pow(n: Int): Float`  
Returns the absolute value of this value.  
`* Special cases:`  
`- NaN.absoluteValue` is `NaN`  
`* @see abs` function  
`* Since Kotlin("1.2")`  
`public expect val Float.absoluteValue: Float`  
Returns the sign of this value:  
`- -1.0` if the value is negative,  
`- zero` if the value is zero,  
`- 1.0` if the value is positive  
`* Special case:`  
`- NaN.sign` is `NaN`  
`* Since Kotlin("1.2")`  
`public expect val Float.sign: Float`  
Returns this value with the sign bit same as of the [sign] value.  
`* If [sign] is NaN the sign of the result is undefined.`  
`* Since Kotlin("1.2")`  
`public expect fun Float.withSign(sign: Float): Float`  
Returns this value with the sign bit same as of the [sign] value.  
`* Since Kotlin("1.2")`  
`public expect fun Float.withSign(sign: Int): Float`  
Rounds this [Float] value to the nearest integer and converts the result to [Int].  
`* Ties are rounded towards positive infinity.`  
`* Special cases:`  
`- x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`  
`- x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`  
`* @throws IllegalArgumentException` when this value is `NaN`  
`* Since Kotlin("1.2")`  
`public expect fun Float.roundToInt(): Int`  
Rounds this [Float] value to the nearest integer and converts the result to [Long].  
`* Ties are rounded towards positive infinity.`  
`* Special cases:`  
`- x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`  
`- x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`  
`* @throws IllegalArgumentException` when this value is `NaN`  
`* Since Kotlin("1.2")`  
`public expect fun Float.roundToLong(): Long`  
// endregion  
// region ===== Integer Math  
===== Returns the absolute value of the given value [n].  
`* Special cases:`  
`- abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow  
`* @see absoluteValue` extension property for [Int]  
`* Since Kotlin("1.2")`  
`public expect fun abs(n: Int): Int`  
Returns the smaller of two values.  
`* Since Kotlin("1.2")`  
`public expect fun min(a: Int, b: Int): Int`  
Returns the greater of two values.  
`* Since Kotlin("1.2")`  
`public expect fun max(a: Int, b: Int): Int`  
Returns the absolute value of this value.  
`* Special cases:`  
`- Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow  
`* @see abs` function  
`* Since Kotlin("1.2")`  
`public expect val`

Int.absoluteValue: Int\n\n\*\*\n \* Returns the sign of this value:\n \* -`-1` if the value is negative,\n \* -`0` if the value is zero,\n \* -`1` if the value is positive\n \*^\n@SinceKotlin("1.2")\n\npublic expect val Int.sign:

Int\n\n\n\*\*\n \* Returns the absolute value of the given value [n].\n \*^\n \* Special cases:\n \* -`abs(Long.MIN\_VALUE)` is `Long.MIN\_VALUE` due to an overflow\n \*^\n \* @see absoluteValue extension property for [Long]\n \*^\n@SinceKotlin("1.2")\n\npublic expect fun abs(n: Long): Long\n\n\*\*\n \* Returns the smaller of two values.\n \*^\n@SinceKotlin("1.2")\n\npublic expect fun min(a: Long, b: Long): Long\n\n\*\*\n \* Returns the greater of two values.\n \*^\n@SinceKotlin("1.2")\n\npublic expect fun max(a: Long, b: Long): Long\n\n\*\*\n \* Returns the absolute value of this value.\n \*^\n \* Special cases:\n \* -`Long.MIN\_VALUE.absoluteValue` is `Long.MIN\_VALUE` due to an overflow\n \*^\n \* @see abs function\n \*^\n@SinceKotlin("1.2")\n\npublic expect val Long.absoluteValue: Long\n\n\*\*\n \* Returns the sign of this value:\n \* -`-1` if the value is negative,\n \* -`0` if the value is zero,\n \* -`1` if the value is positive\n \*^\n@SinceKotlin("1.2")\n\npublic expect val Long.sign: Int\n\n\n// endregion\n","/\*\n \* Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*^\n\npackage kotlin.js\n\n\*\*\n \* Exposes the JavaScript [Math object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global\_Objects/Math) to Kotlin.\n \*^\n@PublishedApi\n\ninternal external object JsMath {\n val LN2: Double\n fun abs(value: Double): Double\n fun acos(value: Double): Double\n fun asin(value: Double): Double\n fun atan(value: Double): Double\n fun atan2(y: Double, x: Double): Double\n fun cos(value: Double): Double\n fun sin(value: Double): Double\n fun exp(value: Double): Double\n fun max(vararg values: Int): Int\n fun max(vararg values: Float): Float\n fun max(vararg values: Double): Double\n fun min(vararg values: Int): Int\n fun min(vararg values: Float): Float\n fun min(vararg values: Double): Double\n fun sqrt(value: Double): Double\n fun tan(value: Double): Double\n fun log(value: Double): Double\n fun cbrt(value: Double): Double\n fun pow(base: Double, exp: Double): Double\n fun round(value: Number): Double\n fun floor(value: Number): Double\n fun ceil(value: Number): Double\n}\n\ninternal const val defineTaylorNBound = `"\n\n var epsilon = 2.220446049250313E-16;\n\n var taylor\_2\_bound = Math.sqrt(epsilon);\n\n var taylor\_n\_bound = Math.sqrt(taylor\_2\_bound);`\n\n\ninternal const val defineUpperTaylor2Bound = `"\n\n\n \$defineTaylorNBound\n\n var upper\_taylor\_2\_bound = 1/taylor\_2\_bound;`\n\n\n\ninternal const val defineUpperTaylorNBound = `"\n\n\n \$defineUpperTaylor2Bound\n\n var upper\_taylor\_n\_bound = 1/taylor\_n\_bound;`\n\n\n\n\n", "names": [], "mappings": "AAWC,CAXA,yB;EACG,IAAI,OAAO,MAAO,KAAI,UAAW,IAAG,MAAM,IAA1C,C;IACI,MAAM,CAAC,QAAD,EAAW,CAAC,SAAD,CAAX,EAAwB,OAAxB,C;SAEL,IAAI,OAAO,OAAQ,KAAI,QAAvB,C;IACD,OAAO,CAAC,MAAM,QAAP,C;;IAGP,IAAI,OAAQ,GAAE,E;IACd,OAAO,CAAC,IAAI,OAAL,C;;CAEd,CAAC,IAAD,EAAO,kB;EACJ,IAAI,IAAI,M;ECPU;;;IAAtB,MAAM,eAAgB,GAAE,a;IACpB,OAAoD,CAA5C,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAe,YAAW,SAAW,KAAG,CAAC,OAAQ,KAAI,c;G;EAGxE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAe,YAAW,SAAU,IAAG,CAAC,OAAQ,KAAI,c;G;EAGID,MAAM,aAAc,GAAE,a;IACiB,OAAO,CAAe,YAAW,U;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAe,YAAW,WAAY,IAAG,CAAC,OAAQ,KAAI,W;G;EAGpD,MAAM,WAAY,GAAE,a;IAChB,OAAO,CAAe,YAAW,U;G;EAGxB,MAAM,aAAc,GAAE,a;IACiB,OAAO,CAAe,YAAW,Y;G;EAGxB,MAAM,cAAe,GAAE,a;IACnB,OAAO,CAAe,YAAW,Y;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAI,W;G;EAG5C,MAAM,QAAS,GAAE,a;IACb,OAAO,KAAK,QA AQ,CAAC,CAAD,CAAI,IAAG,CAAC,CAAC,O;G;EAGjC,MAAM,WAAY,GAAE,a;IACb,OAAO,KAAK,QA AQ,CAAC,CAAD,CAAI,IAAG,WAAW,OAAO,CAAC,CAAD,C;G;EAGjD,MAAM,cAAe,GAAE,a;IACnB,IAAI,CAAE,KAAI,IAAV,C;MAAgB,OAAO,M;IACvB,IAAI,WAAW,MAAM,YAAY,CAAC,CAAD,CAAI,GAAE,MAAM,aAAR,GAAwB,MAAM,S;IACnE,OAAO,GAAl,GAAE,KAAK,UAAU,IAAI,KAAK,CAAC,CAAD,EAAl,a;MAAAc,OAAO,QAAQ,CAAC,CAAD,C;KAAjC,CAAwC,KAAK,CAAC,IAAD,CAAo,GAAE,G;G;EAG/F,MAAM,kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G;EAG5D,MAAM,YAAa,GAAE,gB;IACjB,IAAI,CAAe,KAAI,CAAV,C;MACI,OAAO,I;;IAEX,IAAI,CAAe,KAAI,IAAK,IAAG,CAAe,KA

AI,IAAK,IAAG,CAAC,MAAM,WAAW,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAL,CAAC,OAAvE,C;MA  
CI,OAAO,K;;IAGX,KAAK,IAAI,IAAI,CAAR,EA AW,IAAI,CAAC,OAArB,EAA8B,CAAE,GAAE,CAAIC,EAAq  
C,CAAC,EAAtC,C;MACI,IAAI,CAAC,MAAM,OAAO,CAAC,CAAC,CAAC,CAAD,CAAF,EAAO,CAAC,CAA  
C,CAAD,CAAR,CAAIB,C;QACI,OAAO,K;;IAGf,OAAO,I;G;EAGX,MAAM,gBAAiB,GAAE,gB;IACrB,OAAO,  
MAAM,OAAO,YAAY,sBAAsB,CAAC,CAAD,EA AI,CAAJ,C;G;EAG1D,MAAM,cAAe,GAAE,e;IACnB,IAAI,G  
AAI,KAAL,IAAZ,C;MAAkB,OAAO,C;IACzB,IAAI,SAAS,C;IACb,KAAK,IAAI,IAAI,CAAR,EA AW,IAAI,GAA  
G,OAAvB,EAAgC,CAAE,GAAE,CAApC,EAAuC,CAAC,EA AxC,C;MACI,MAAO,GAAqB,CAAjB,EAAG,GAA  
E,MAAO,GAAE,CAAG,IAAE,MAAM,SAAS,CAAC,GAAG,CAAC,CAAD,CAAJ,CAAU,GAAE,C;;IAE7D,OA  
AO,M;G;EAGX,MAAM,kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G  
;EAG5D,MAAM,mBAAoB,GAAE,iB;IACxB,KAAK,KAAK,CAAC,MAAM,gBAAP,C;G;ECpFQ;;;IAAtB,MAA  
M,eAAgB,GAAE,mB;IACpB,CAAC,aAAc,GAAE,I;IACjB,OAAO,C;G;EAGX,MAAM,uBAAwB,GAAE,4C;IAC  
5B,MAAM,IAAK,GAAE,M;IACb,MAAM,IAAK,GAAE,M;IACb,MAAM,aAAc,GAAE,I;IACtB,OAAO,mBAAm  
B,CAAC,MAAD,EAAS,MAAT,EA AiB,6BAA6B,CAAC,UAAD,CAA9C,C;G;EAG9B,iD;IACI,GAAG,WAA Y,G  
AAE,sBAAsB,CAAC,OAAO,MAAO,KAAL,UAAW,GAAE,KAAK,QAAP,GA AkB,KAAK,UAArD,C;IACvC,GA  
AG,YAAa,GAAE,G;IACIB,OAAO,G;G;EAGX,IAAI,gCAAgC,CACHC,CACI,OADJ,EACa,CAAE,KAAF,EAAS,  
IAAT,EA Ae,oBAAf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,kB;GADvB,CADb,EAI,SAJJ,EAIe,CAAE  
,KAAF,EAAS,IAAT,EA Ae,oBAAf,EAAqC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CADgC,E  
AShC,CACI,OADJ,EACa,CAAE,KAAF,EAAS,IAAT,EA Ae,oBAAf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,Q  
AAQ,kB;GADvB,CADb,EAI,SAJJ,EAIe,CAAE,KAAF,EAAS,IAAT,EA Ae,oBAAf,EAAqC,Y;IAC5C,OAAO,M  
AAM,OAAO,QAAQ,W;GADrB,CAJf,CATgC,C;EAmBpC,uC;IACI,IAAI,KAAK,MAAO,KAAL,IAApB,C;MACI,  
KAAK,MAAO,GAAE,CACV,UADU,EACE,CAAC,KAAK,qBAAqB,EAA3B,CADF,EA EV,SAFU,EAEC,IAFD,  
EAGV,SAHU,EAGC,EAHD,EAIV,UAJU,EAIE,EAJF,EAKV,KALU,EAKH,EALG,EAMV,aANU,EAMK,EANL,  
C;;IASIB,OAAO,KAAK,M;G;EChDD;;;IAAf,MAAM,QAAS,GAAE,a;IACb,OAAoB,CAAZ,CAAE,GAAE,KAA  
Q,KAAG,EAAG,IAAG,E;G;EAGjC,MAAM,OAAQ,GAAE,a;IACZ,OAAkB,CAAV,CAAE,GAAE,GAAM,KAA  
G,EAAG,IAAG,E;G;EAG/B,MAAM,OAAQ,GAAE,a;IACZ,OAAO,CAAE,GAAE,K;G;EAGf,MAAM,aAAc,GA  
AE,a;IACIB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAF,GAAM,MAAM,KAAK,WAAW,CAAC,CAAD  
,C;G;EAGhE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAC,MAAM,EA  
T,GAAc,MAAM,YAAY,CAAC,CAAD,C;G;EAGpE,MAAM,cAAe,GAAE,a;IACnB,OAAO,MAAM,QAAQ,CAA  
C,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGzB,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CA  
AC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,MAAM,eAAgB,GAAE,a;IACpB,OAAO,CAAC,C;G;EA  
GZ,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EA  
GxB,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,GAAE,UAA R,C;MAAoB,OAAO,U;IAC3B,IAAI,CAAE,GAAE  
,WAAR,C;MAAqB,OAAO,W;IAC5B,OAAO,CAAE,GAAE,C;G;EAGf,MAAM,YAAa,GAAE,a;IACjB,IAAI,CA  
AE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,IAAI,CAAE,YAAW,MAAM,UAAvB,C;MAAmC,OAAO,C;IAC1C,O  
AAO,IAAI,MAAM,UAAV,CAAqB,CAArB,C;G;EAGX,MAAM,UAAW,GAAE,a;IACf,IAAI,CAAE,IAAG,IAAT  
,C;MAAe,OAAO,C;IACtB,OAAO,MAAM,OAAO,CAAC,CAAD,C;G;ECIDV;;;IAAd,MAAM,OAAQ,GAAE,sB;I  
ACZ,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,IAAK,IAAG,I;;IAGnB,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,  
K;;IAGX,IAAI,IAAK,KAAL,IAAb,C;MACI,OAAO,IAAK,KAAL,I;;IAGpB,IAAI,OAAO,IAAK,KAAL,QAAS,IAA  
G,OAAO,IAAI,OAAQ,KAAL,UAAvD,C;MACI,OAAO,IAAI,OAAO,CAAC,IAAD,C;;IAGtB,IAAI,OAAO,IAAK,  
KAAL,QAAS,IAAG,OAAO,IAAK,KAAL,QA AhD,C;MACI,OAAO,IAAK,KAAL,IAAK,KAAL,IAAK,KAAL,CAA  
E,IAAG,CAAE,GAAE,IAAK,KAAL,CAAE,GAAE,IAAnC,C;;IAGzB,OAAO,IAAK,KAAL,I;G;EAGpB,MAAM,S  
AAU,GAAE,e;IACd,IAAI,GAAL,IAAG,IAAX,C;MACI,OAAO,C;;IAEX,IAAI,UAAU,OAAO,G;IACrB,IAAI,QA  
AS,KAAL,OAAjB,C;MACI,OAAO,UAAW,KAAL,OAAO,GAAG,SAAU,GAAE,GAAG,SAAS,EAAd,GA AmB,iB  
AAiB,CAAC,GAAD,C;;IAEIF,IAAI,UAAW,KAAL,OAA nB,C;MACI,OAAO,iBAAiB,CAAC,GAAD,C;;IAE5B,I  
AAI,QAAS,KAAL,OAAjB,C;MACI,OAAO,MAAM,eAAe,CAAC,GAAD,C;;IAEhC,IAAI,SAAU,KAAL,OAAIB,C  
;MACI,OAAO,MAAM,CAAC,GAAD,C;;IAGjB,IAAI,MAAM,MAAM,CAAC,GAAD,C;IACbB,OAAO,iBAAiB,  
CAAC,GAAD,C;G;EAI5B,MAAM,SAAU,GAAE,a;IACd,IAAI,CAAE,IAAG,IAAT,C;MACI,OAAO,M;WAEN,I  
AAI,MAAM,WAAW,CAAC,CAAD,CAArB,C;MACD,OAAO,O;;MAGP,OAAO,CAAC,SAAS,E;;GART,Y;EA Ah



AK,UAAU,SAAU,GAAE,qB;IAC/B,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE,CAAE,IAAG,EA  
AG,GAAE,KAAiB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAAiB,C;:IAGb,IAAI,IAAI,OAAO,EAaf,  
C;MACE,OAAO,G;:IAGT,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UA  
AZ,CAAnB,C;QAGE;:YAAI,YAAAY,MAAM,KAAK,WAAW,CAAC,KAAD,C;QACtC,IAAI,MAAM,IAAI,IAAI,  
CAAC,SAAD,C;QACiB,IAAI,MAAM,GAAG,SAAS,CAAC,SAAD,CAAW,SAAS,CAAC,IAAD,C;QACiC,OAA  
O,GAAG,SAAS,CAAC,KAAD,CAAQ,GAAE,GAAG,MAAM,EAAE,SAAS,CAAC,KAAD,C;:QAEjD,OAAO,G  
AAI,GAAE,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,C;:;:IAMvC,IAAI,eAAe,MAAM,KAAK,WAAW,CAAC,I  
AAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,MAAM,I;IACV,IAAI,SAAS,E;IACb,OAAO,IAAP  
,C;MACE,IAAI,SAAS,GAAG,IAAI,CAAC,YAAD,C;MACpB,IAAI,SAAS,GAAG,SAAS,CAAC,MAAM,SAAS,  
CAAC,YAAD,CAAhB,CAA+B,MAAM,E;MAC9D,IAAI,SAAS,MAAM,SAAS,CAAC,KAAD,C;MAE5B,GAAL,  
GAAE,M;MACN,IAAI,GAAG,OAAO,EAAd,C;QACE,OAAO,MAAO,GAAE,M;:QAEhB,OAAO,MAAM,OAAQ  
,GAAE,CAAvB,C;UACE,MAAO,GAAE,GAAL,GAAE,M;:QAEjB,MAAO,GAAE,EAAG,GAAE,MAAO,GAAE,  
M;:;GAzCE,0D;EAgd/B,MAAM,KAAK,UAAU,YAAa,GAAE,Y;IACiC,OAAO,IAAI,M;GADqB,yD;EAMiC,MA  
AM,KAAK,UAAU,WAAAY,GAAE,Y;IACjC,OAAO,IAAI,K;GADoB,4D;EAMjC,MAAM,KAAK,UAAU,mBAAo  
B,GAAE,Y;IACzC,OAAQ,IAAI,KAAM,IAAG,CAAG,GACpB,IAAI,KADgB,GACR,MAAM,KAAK,gBAAiB,G  
AAE,IAAI,K;GAFX;:;I;EAUzC,MAAM,KAAK,UAAU,cAAe,GAAE,Y;IACpC,IAAI,IAAI,WAAW,EAAnB,C;M  
ACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;QACE,OAAO,E;:QAEp,OAAO,IAAI,OAAO,  
EAAE,cAAc,E;:;MAGpC,IAAI,MAAM,IAAI,MAAO,IAAG,CAAE,GAAE,IAAI,MAAN,GAAe,IAAI,K;MAC7C,  
KAAK,IAAI,MAAM,EAaf,EAAmB,GAAL,GAAE,CAAzB,EA4B,GAAG,EAA/B,C;QACE,IAAuB,CAAIB,GA  
AI,GAAG,CAAE,IAAG,GAAM,KAAG,CAAiB,C;UACE,K;:;MAGJ,OAAO,IAAI,MAAO,IAAG,CAAE,GAAE,  
GAAL,GAAE,EAAR,GAAa,GAAL,GAAE,C;:GAdV,mD;EAoBpC,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7  
B,OAAO,IAAI,MAAO,IAAG,CAAE,IAAG,IAAI,KAAM,IAAG,C;GADZ,uD;EAM7B,MAAM,KAAK,UAAU,W  
AAAY,GAAE,Y;IACjC,OAAO,IAAI,MAAO,GAAE,C;GADW,kD;EAMjC,MAAM,KAAK,UAAU,MAAO,GAAE,  
Y;IAC5B,OAAuB,CAAf,IAAI,KAAM,GAAE,CAAG,KAAG,C;GADA;:;I;EAS5B,MAAM,KAAK,UAAU,WAAAY  
,GAAE,iB;IACjC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAAM,IAAG,KAAK,K;GADiB;:;I;EA  
SjC,MAAM,KAAK,UAAU,cAAe,GAAE,iB;IACpC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAA  
M,IAAG,KAAK,K;GADvB;:;I;EASpC,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,QAAQ,CAA  
C,KAAD,CAAQ,GAAE,C;GADA;:;I;EAS/B,MAAM,KAAK,UAAU,gBAAiB,GAAE,iB;IACtC,OAAO,IAAI,QA  
AQ,CAAC,KAAD,CAAQ,IAAG,C;GADM;:;I;EAStC,MAAM,KAAK,UAAU,YAAa,GAAE,iB;IACiC,OAAO,IA  
AI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;GADG;:;I;EASIC,MAAM,KAAK,UAAU,mBAAoB,GAAE,iB;IACzC,  
OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;GADS;:;:;I;EAWzC,MAAM,KAAK,UAAU,QAAS,GAAE,iB  
;IAC9B,IAAI,IAAI,WAAW,CAAC,KAAD,CAAnB,C;MACE,OAAO,C;:IAGT,IAAI,UAAU,IAAI,WAAW,E;IAC  
7B,IAAI,WAAW,KAAK,WAAW,E;IAC/B,IAAI,OAAQ,IAAG,CAAC,QAAhB,C;MACE,OAAO,E;:IAET,IAAI,C  
AAC,OAAQ,IAAG,QAAhB,C;MACE,OAAO,C;:;IAIT,IAAI,IAAI,SAAS,CAAC,KAAD,CAAO,WAAW,EAAnC,  
C;MACE,OAAO,E;:MAEP,OAAO,C;:GAIbMb,wD;EAWb9B,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,I  
AAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,MAAM,KAAK,U;:MAEiB,OAAO,I  
AAI,IAAI,EAAE,IAAI,CAAC,MAAM,KAAK,IAAZ,C;:GAJl;:;:;I;EAc7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB  
;IAGiB;QAAI,MAAM,IAAI,MAAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,  
IAAI,KAAM,KAAI,E;IACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MAAO,KAAI,E;  
IACiB,IAAI,MAAM,KAAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB,IAAI,MAA  
M,KAAK,KAAM,GAAE,K;IAEvB,IAAI,MAAM,CAAV,EAAa,MAAM,CAAnB,EAA5B,MAAM,CAA5B,EAA+  
B,MAAM,C;IACrC,GAAL,IAAG,GAAL,GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,  
GAAL,IAAG,GAAL,GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,GAAL,IAAG,GAAL,  
GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,GAAL,IAAG,GAAL,GAAE,G;IACb,GA  
L,IAAG,K;IACP,OAAO,MAAM,KAAK,SAAS,CAAE,GAAL,IAAG,EAAL,GAAE,GAaf,EAaqB,GAAL,IAAG,EA  
AL,GAAE,GAAL,C;GAzBH;:;:;I;EAKc1B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,IAAI,CA  
AC,KAAK,OAAO,EAAb,C;GADc;:;:;I;EAU/B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,IAAI,IAAI,OAAO  
,EAaf,C;MACE,OAAO,MAAM,KAAK,K;WACb,IAAI,KAAK,OAAO,EAAhB,C;MACL,OAAO,MAAM,KAAK,





GAAE,KAAK,MADtB,C;GADJ;;;I;EAWzB,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KA  
AK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAIB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;GADH;;;I;  
EAW1B,MAAM,KAAK,UAAU,UAAW,GAAE,mB;IAChC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;  
MACE,OAAO,I;;MAEP,IAAI,MAAM,IAAI,K;MACd,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,OAAO,IAAI,M;  
QACf,OAAO,MAAM,KAAK,SAAS,CACvB,GAAl,IAAG,OADgB,EAeTB,IAAK,IAAG,OAAS,GAAG,GAAl,KA  
AK,EAAG,GAAE,OAFZ,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAD,EAAl,GAAl,IAAI,OAAQ,GAA  
E,EAAtB,C;;GAZD;;;I;EAuBhC,MAAM,KAAK,UAAU,WAAY,GAAE,mB;IACjC,OAAQ,IAAG,E;IACX,IAAI,  
OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE  
,IAAI,MAAM,IAAI,K;QACd,OAAO,MAAM,KAAK,SAAS,CACtB,GAAl,KAAl,OAAS,GAAG,IAAK,IAAI,EA  
AG,GAAE,OADZ,EAeVB,IAAK,IAAG,OAFe,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CACvB,IAAK,IAAI,OA  
AQ,GAAE,EADI,EAeVB,IAAK,IAAG,CAAE,GAAE,CAAF,GAAM,EAFO,C;;GAZA;;;;;I;EA2BjC,MAAM,KA  
AK,UAAU,mBAaOB,GAAE,mB;IACzC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;M  
AEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,MAA  
M,KAAK,SAAS,CACtB,GAAl,KAAl,OAAS,GAAG,IAAK,IAAI,EAAG,GAAE,OADZ,EAeVB,IAAK,KAAl,OA  
Fc,C;aAGtB,IAAI,OAAQ,IAAG,EAaf,C;QACL,OAAO,MAAM,KAAK,SAAS,CAAC,IAAD,EAAO,CAAP,C;;Q  
AE3B,OAAO,MAAM,KAAK,SAAS,CAAC,IAAK,KAAK,OAAQ,GAAE,EAArB,EAA0B,CAA1B,C;;GAdQ;A,E  
AoBzC,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC3B,OAAO,KAAM,YAAW,MAAM,KAAM,IAAG,IAAI,W  
AAW,CAAC,KAAD,C;G;EAG1D,MAAM,KAAK,UAAU,gBAaiB,GAAE,MAAM,KAAK,UAAU,Q;EAE7D,MA  
AM,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,IAAZ,C;G;EAGnB,MAA  
M,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,QAAs,C;G;EAGnB,MAAM,  
KAAK,UAAU,QAAS,GAAE,Y;IAC5B,OAAO,IAAI,SAAS,E;G;EAGxB,MAAM,KAAK,UAAU,UAAW,GAAE,  
Y;IAC9B,OAAO,I;G;EAGX,MAAM,KAAK,UAAU,WAAY,GAAE,MAAM,KAAK,UAAU,O;EACxD,MAAM,K  
AAK,UAAU,IAAK,GAAE,MAAM,KAAK,UAAU,I;EAEjD,MAAM,KAAK,UAAU,QAAS,GAAE,iB;IAC5B,OA  
AO,IAAI,MAAM,OAAO,OAAO,UAAxB,CAAmC,IAAnC,EAAYC,KAAzC,C;G;EC1zBS;;;;;IAApB,MAAM,aA  
Ac,GAAE,2B;G;EAGtB,MAAM,qBAAsB,GAAE,oB;IAC1B,OAAO,G;G;EAGX,MAAM,aAAc,GAAE,e;IACIB,I  
AAI,IAAI,Y;MACJ,CAAE,GAAE,GAAG,E;MACP,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;IAEI  
B,OAAO,Y;MACH,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;G;EAItB,MAAM,SAAU,GAAE,gB;I  
ACd,OAAO,kB;MACH,OAAO,OAAO,MAAO,KAAl,I;K;G;EAIjC,MAAM,aAAc,GAAE,iB;IACIB,OAAO,kB;M  
ACH,OAAO,MAAM,OAAO,CAAC,MAAD,EAAS,KAAT,C;K;G;EAI5B,MAAM,OAAQ,GAAE,c;IACZ,OAAO,  
kB;MACH,OAAO,MAAO,IAAG,IAAK,IAAG,EAAE,CAAC,MAAD,C;K;G;EAIInC,MAAM,aAAc,GAAE,gB;IA  
CIB,OAAO,kB;MACH,OAAO,CAAC,CAAC,MAAD,CAAS,IAAG,CAAC,CAAC,MAAD,C;K;G;EAI7B,MAAM,  
qBAAsB,GAAE,wC;G;EAG9B,MAAM,YAAa,GAAE,iB;IACjB,OAAO,K;G;EAGX,MAAM,gBAaiB,GAAE,qB;  
IACrB,gBAAgB,E;G;EAGpB,MAAM,oBAaqB,GAAE,qB;IACzB,gBAAgB,E;G;EAGpB,MAAM,kBAAmB,GAA  
E,qB;IACvB,gBAAgB,E;G;EAGpB,MAAM,mBAaOB,GAAE,4B;IACxB,gBAAgB,E;G;EAGpB,MAAM,6BAA8  
B,GAAE,yB;IACIC,gBAAgB,E;G;EAGpB,4B;IACI,MAAM,IAAI,KAAJ,CACf,iDAakD,GACID,qDAAsD,GACt  
D,uDAHE,C;G;EAMV,MAAM,gBAaiB,GAAE,4B;IACrB,OAAO,Y;MACH,OAAO,Y;K;G;ECjFE;;;IAAjB,MAA  
M,UAAW,GAAE,gB;IACf,IAAI,QAAQ,OAAO,C;IACnB,IAAI,KAAM,KAAl,QAAd,C;MACI,IAAI,OAAO,CAA  
E,KAAl,QAajB,C;QACI,OAAO,MAAM,gBAAgB,CAAC,CAAD,EAAl,CAAJ,C;;MAEjC,OAAO,MAAM,mBA  
AmB,CAAC,CAAD,EAAl,CAAJ,C;;IAEpC,IAAI,KAAM,KAAl,QAAS,IAAG,KAAM,KAAl,SAApC,C;MACI,O  
AAO,MAAM,mBAAmB,CAAC,CAAD,EAAl,CAAJ,C;;IAEpC,OAAO,CAAC,gBAAgB,CAAC,CAAD,C;G;EAG  
5B,MAAM,mBAaOB,GAAE,gB;IACxB,OAAO,CAAE,GAAE,CAAE,GAAE,EAaf,GAAO,CAAE,GAAE,CAAE  
,GAAE,CAAF,GAAM,C;G;EAGpC,MAAM,gBAaiB,GAAE,gB;IACrB,IAAI,CAAE,GAAE,CAAR,C;MAAW,O  
AAO,E;IACIB,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,C;IAEIB,IAAI,CAAE,KAAl,CAAV,C;MACI,IAAI,  
CAAE,KAAl,CAAV,C;QAAa,OAAO,C;MAEpB,IAAI,KAAK,CAAE,GAAE,C;MACb,OAAO,EAAG,KAAl,CA  
AE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EAaf,GAAO,C;;IAG7C,OAAO,CAAE,KA  
Al,CAAE,GAAG,CAAE,KAAl,CAAE,GAAE,CAAF,GAAM,CAAJB,GAAsB,E;G;EAGzC,MAAM,QAAS,GAAE  
,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,QAAS,GAAE,iB;IACb,OA  
O,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,KAAM,GAAE,IAAI,KAAM,IAAG,I;EAE3

B,MAAM,aAAc,GAAE,I;EAETb,oB;IACI,OAAyB,CAAhB,CAAE,GAAE,YAAy,KAAG,CAAE,GAAE,KAAP,C  
AAe,GAAe,CAAZ,CAAE,GAAE,KAAQ,KAAG,CAAE,GAAE,CAAP,CAAW,GAAE,C;G;EA6DtE,CA1DD,Y;IA  
CG,IAAI,MAAM,IAAI,WAAJ,CAAgB,CAAhB,C;IACV,IAAI,aAAa,IAAI,YAAJ,CAAIb,GAAjB,C;IACjB,IAAI,  
aAAa,IAAI,YAAJ,CAAIb,GAAjB,C;IACjB,IAAI,WAAW,IAAI,UAAJ,CAAE,GAAf,C;IACf,IAAI,WAAW,C;IAC  
f,IAAI,YAAy,C;IAEhB,UAAU,CAAC,CAAD,CAAI,GAAE,EAaf;A,IACd,IAAI,QAAQ,CAAC,QAAD,CAAW,  
KAAI,CAA3B,C;MACI,QAAS,GAAE,C;MACX,SAAU,GAAE,C;;IAGhB,MAAM,aAAc,GAAE,iB;MACIB,OAA  
O,MAAM,gBAAgB,CAAC,KAAK,CAAC,KAAD,CAAQ,GAAE,GAAf,GAAQ,KAAtB,C;K;IAGjC,MAAM,gBA  
AiB,GAAE,iB;MACrB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,MAAM,KAAK,SAAS,CAAC,QA  
AQ,CAAC,QAAD,CAAT,EAAqB,QAAQ,CAAC,SAAD,CAA7B,C;K;IAG/B,MAAM,eAAgB,GAAE,iB;MACpB,  
QAAQ,CAAC,QAAD,CAAW,GAAE,KAAK,K;MAC1B,QAAQ,CAAC,SAAD,CAAY,GAAE,KAAK,M;MAC3B,  
OAAO,UAAU,CAAC,CAAD,C;K;IAGrB,MAAM,YAAa,GAAE,iB;MACjB,OAAO,MAAM,eAAe,CAAC,KAAK,  
CAAC,KAAD,CAAQ,GAAE,GAAf,GAAQ,KAAtB,C;K;IAGhC,MAAM,eAAgB,GAAE,iB;MACpB,UAAU,CA  
AC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,CAAD,C;K;IAGnB,MAAM,cAAe,GAAE,iB;MACnB,  
QAAQ,CAAC,CAAD,CAAI,GAAE,K;MACd,OAAO,UAAU,CAAC,CAAD,C;KAFa;A,IAMrB,MAAM,cAAe,G  
AAE,iB;MACnB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,SAAD,CAAY,GAAE,a;K  
;IAGjC,MAAM,eAAgB,GAAE,e;MACpB,IAAc,CAAT,GAAl,GAAE,CAAG,MAAI,GAAlB,C;QACI,OAAO,GA  
Al,GAAE,C;;QAGb,UAAU,CAAC,CAAD,CAAI,GAAE,G;QACHb,OAAc,CAA9B,QAAQ,CAAC,SAAD,CAA  
Y,GAAE,EAAG,GAAE,CAAG,IAAE,QAAQ,CAAC,QAAD,CAAW,GAAE,C;;K;GAGvE,G;EAef,MAAM,cAAe  
,GAAE,a;IACnB,OAAO,CAAE,IAAG,IAAK,GAAE,CAAF,GAAM,MAAM,SAAS,E;G;EC7G1C;;;QAAI,OAAO,  
MAAM,UAAU,WAAy,KAAI,WAA3C,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAp,EAAMb,YAAAnB,EAAlC,C  
ACID,KADkD,EAC3C,kC;MACH,QAAS,GAAE,QAAS,IAAG,C;MACvB,OAAO,IAAI,YAAy,CAAC,YAAD,E  
AAe,QAaf,CAAyB,KAAI,Q;KAHN,CAAjC,C;;EAozB,IAAI,OAAO,MAAM,UAAU,SAAU,KAAI,WAAzC,C;I  
ACI,MAAM,eAAe,CAAC,MAAM,UAAp,EAAMb,UAAAnB,EAAB+B,CACHd,KADgD,EACzC,kC;MACH,IAAI,g  
BAAgB,IAAI,SAAS,E;MACjC,IAAI,QAAS,KAAI,SAAU,IAAG,QAAS,GAAE,aAAa,OAAtD,C;QACI,QAAS,G  
AAE,aAAa,O;;MAE5B,QAAS,IAAG,YAAy,O;MACxB,IAAI,YAAy,aAAa,QAAQ,CAAC,YAAD,EAae,QAaf,  
C;MACrC,OAAO,SAAU,KAAI,EAAG,IAAG,SAAU,KAAI,Q;KARG,CAAB,C;;EAazB,IAAI,OAAO,IAAI,KA  
AM,KAAI,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,CAAE,GAAE,CAAC,CAAH;A,MACF,IAAI,CAAE,K  
AAI,CAAE,IAAG,KAAK,CAAC,CAAD,CAApB,C;QACI,OAAO,MAAM,CAAC,CAAD,C;;MAEjB,OAAO,CAA  
E,GAAE,CAAE,GAAE,CAAF,GAAM,E;K;;EAG3B,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MA  
AO,GAAE,a;MACT,IAAI,KAAK,CAAC,CAAD,CAAT,C;QACI,OAAO,G;;MAEX,IAAI,CAAE,GAAE,CAAR,C;  
QACI,OAAO,IAAI,MAAM,CAAC,CAAD,C;;MAErB,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAuKtB,CAnKD,  
Y;IACG,IAAI,UAAU,qB;IACd,IAAI,iBAAiB,IAAI,KAAK,CAAC,OAAD,C;IAC9B,IAAI,iBAAiB,IAAI,KAAK,  
CAAC,cAAD,C;IAC9B,IAAI,uBAAuB,CAAC,GAAC,c;IAC7B,IAAI,uBAAuB,CAAC,GAAC,c;IAE7B,IAAI,OA  
AO,IAAI,KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,G  
AAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO  
,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAEP,IAAI,IAAI,IAAI,IAAI,CAAC,CA  
AD,C;UACHb,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAb,C;YAAkB,OAAO,  
IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,IAAT,C;UACjC,IAAI,CAAC,QAAQ,CAAC,EAAD,CAAb,C;YAAmB,OA  
AO,CAAC,IAAI,IAAI,CAAC,CAAC,CAAE,GAAE,IAAI,IAAV,C;UACnC,OAAgB,CAAR,CAAE,GAAE,EAAl,I  
AAE,C;;O;;IAI9B,IAAI,OAAO,IAAI,KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,I  
AAI,IAAI,CAAC,CAAD,C;QACHb,IAAI,KAAK,CAAE,GAAE,C;QACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CA  
Al,IAAG,CAAC,QAAQ,CAAC,EAAD,CAA7B,C;UAAmC,OAAO,IAAI,IAAI,CAAC,IAAI,IAAI,CAAC,CAAD,  
CAAI,GAAE,IAAI,IAAnB,C;QACID,OAAgB,CAAR,CAAE,GAAE,EAAl,IAAE,C;;O;;IAI1B,IAAI,OAAO,IAAI,  
KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cA  
AIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,C  
AAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAGP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAC,CA  
AF,CAAhB,EAASB,IAAI,IAAI,IAAI,CAAC,CAAC,CAAF,C;UACIC,OAAO,CAAE,KAAI,QAAS,GAAE,CAAF,  
GAAM,CAAE,KAAI,QAAS,GAAE,EAaf,GAAe,CAAP,CAAE,GAAE,CAAG,KAAg,CAAE,GAAE,CAAP,C;;O

;;;IAQtE,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,QAAQ,a;QACR,IAAI,CAAE,IAAG,CAAC,c  
AAV,C;UAEI,IAAI,CAAE,GAAE,oBAAR,C;YAEI,IAAI,CAAE,GAAE,oBAAR,C;cAGI;qBAAO,IAAI,IAAI,CA  
AC,CAAD,CAAI,GAAE,IAAI,I;;cAKzB;qBAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAE,GAAG,CAAE,IAAG,C  
AAE,GAAE,CAAP,CAAZ,C;;;YAKnB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GA  
AE,CAAE,GAAE,CAAT,CAAd,C;;eAGIB,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAED,OAAO,CAAC,KAAK,CA  
AC,CAAC,CAAF,C;;UAKb;cAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,IAAG,cAAmB,C;YAEI,I  
AAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,C;;UAEnB,OAAO,M;;O  
;MAGf,IAAI,MAAO,GAAE,K;;IAEjB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a  
;QACT,IAAI,CAAE,GAAE,CAAR,C;UAEI,OAAO,G;eAEN,IAAI,CAAE,GAAE,CAAE,IAAG,cAAb,C;UAED,I  
AAI,CAAE,GAAE,oBAAR,C;YAGI;mBAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,I;;YAIzB,OAAO,IA  
AI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CAAT,CAAd,C;;;UAKnB,IAAI,  
IAAI,IAAI,KAAK,CAAC,CAAE,GAAE,CAAL,CAAjB;A,UAEA,IAAI,SAAS,C;UACb,IAAI,CAAE,IAAG,cAAT  
,C;YAEI,IAAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,E;;UAGnB,O  
AAO,IAAI,KAAK,CAAC,CAAD,CAAI,GAAE,M;;O;;IAIIC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,  
IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb  
,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,  
GAAE,C;;UAE5B,OAAO,M;;QAEX,OAAO,IAAI,IAAI,CAAS,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAA  
E,CAAP,CAAT,CAAoB,GAAE,C;O;;IAG7C,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,G  
AAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb  
,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,CAAC,EAAG,GAAE,  
CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,C  
AAE,GAAE,CAAL,C;O;;IAGvB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QAC  
T,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAA  
K,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,EAAG,GAAE,EAAG,GAAE,EA  
G,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,  
C;O;;GAG/B,G;EACF,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,Y;MACT,IAAI,IA  
AI,C;MACR,IAAI,SAAS,SAAS,O;MAEtB,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAApB,EAA4B,CA  
AC,EAA7B,C;QACI,IAAI,SAAS,CAAC,CAAD,CAAI,KAAI,QAAS,IAAG,SAAS,CAAC,CAAD,CAAI,KAAI,C  
AAC,QAAnD,C;UACI,OAAO,Q;;QAEX,CAAE,IAAG,SAAS,CAAC,CAAD,CAAI,GAAE,SAAS,CAAC,CAAD,  
C;;MAEjC,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAGxB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,I  
AAI,MAAO,GAAE,a;MACT,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,O;K;;EAGjC,IAAI,OAAO,IA  
AI,KAAI,KAAM,KAAI,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,  
IAAI,M;K;;EAGjC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAG,oB;MACV,OAAO,a;  
QACH,IAAI,SAAS,CAAE,KAAI,C;QACnB,IAAI,MAAO,KAAI,CAAF,C;UACI,OAAO,E;;QAEX,OAAO,EAAG,  
IAAG,GAAG,CAAC,MAAD,CAAS,GAAE,GAAL,GAAE,CAAvB,CAA0B,GAAE,CAAtC;A,O;KAEN,CAAC,IA  
AI,IAAL,EAAW,IAAI,IAAf,C;;EAIN,IAAI,OAAO,WAAW,OAAQ,KAAI,WAAIC,C;IACI,WAAW,OAAQ,GAA  
E,a;MACjB,OAAO,CAAE,IAAG,IAAK,IAAG,CAAC,UAAW,IAAG,IAAK,IAAG,CAAC,UAAU,UAAW,KAAI,  
SAAS,UAAU,U;K;;EAIhG,IAAI,OAAO,KAAK,UAAU,KAAM,KAAI,WAApC,C;IAEyB;IAArB,MAAM,eAAe,C  
AAC,KAAK,UAAU,EAakB,MAAIB,EAA0B,CAC3C,KAD2C,EACpC,iB;MAGH;UAALI,IAAK,IAAG,IAAZ,C;Q  
ACI,MAAM,IAAI,SAAJ,CAAc,6BAAd,C;;MAGV,IAAI,IAAI,MAAM,CAAC,IAAD,CAAd;A,MAGA,IAAI,MA  
AM,CAAC,OAAQ,KAAI,CAAvB;A,MAGA,IAAI,QAAQ,SAAS,CAAC,CAAD,C;MACrB,IAAI,gBAAgB,KAA  
M,IAAG,CAA7B;A,MAGA,IAAI,IAAI,aAAc.GAAE,CAAE,GACIB,IAAI,IAAI,CAAC,GAAL,GAAE,aAAP,EAA  
sB,CAAtB,CADU,GAElB,IAAI,IAAI,CAAC,aAAD,EAAGB,GAAhB,CAFhB;A,MAKA,IAAI,MAAM,SAAS,CA  
AC,CAAD,C;MACnB,IAAI,cAAc,GAAL,KAAI,SAAU,GACIB,GADkB,GACZ,GAAL,IAAG,CAD/B;A,MAIA,IA  
AI,aAAa,WAAY,GAAE,CAAE,GACHB,IAAI,IAAI,CAAC,GAAL,GAAE,WAAP,EAAoB,CAApB,CADQ,GAElB  
,IAAI,IAAI,CAAC,WAAD,EAAC,GAAd,CAFzB;A,MAKA,OAAO,CAAE,GAAE,UAAx,C;QACI,CAAC,CAAC,  
CAAD,CAAI,GAAE,K;QACP,CAAC,E;;;MAIL,OAAO,C;KAvCgC,CAA1B,C;;EA4HvB,CAhFD,Y;IACG,yC;M  
ACI,IAAI,MAAO,GAAE,CAAb,C;QAAGB,OAAO,IAAI,IAAI,CAAC,CAAD,EAAL,MAAO,GAAE,MAAb,C;MA

C/B,OAAO,IAAI,IAAI,CAAC,MAAD,EAAS,MAAT,C;K;IAEnB,qC;MACI,IAAI,OAAO,GAAL,KAAL,WAAAB,C;QACI,GAAL,GAAE,IAAI,O;;MAEd,KAAM,GAAE,eAAe,CAAC,KAAM,IAAG,CAAV,EAAa,IAAI,OAAjB,C;MACvB,GAAL,GAAE,IAAI,IAAI,CAAC,KAAD,EAAQ,eAAe,CAAC,GAAD,EAAM,IAAI,OAAV,CAAvB,C;MACd,OAAO,IAAI,IAAI,YAAR,CAAqB,IAAI,SAAS,CAAC,KAAD,EAAQ,GAAR,CAAIC,C;K;IAGX,IAAI,SAAS,CAAC,SAAD,EAAY,UAAZ,EAAwB,WAAxB,EAAqC,UAArC,EAAiD,YAAjD,EAA+D,YAA/D,C;IACb,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAL,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAAX,EAAuB,MAAvB,EAA+B,CAChD,KADgD,EACzC,KAAK,UAAU,KAD0B,CAA/B,C;;MAIzB,IAAI,OAAO,UAAU,UAAU,MAAO,KAAL,WAA1C,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAAX,EAAuB,OAAvB,EAAGC,CACjD,KADiD,EAC1C,eAD0C,CAAhC,C;;;MAQJ,CAApB,Y;OAAc,MAAM,CAAC,IAAD,EAAO,IAAI,UAAJ,CAAe,CAAf,CAAP,E;;MAErB,IAAI,QAAQ,QAAQ,UAAU,M;MAC9B,MAAM,eAAe,CAAC,QAAQ,UAAAT,EAAqB,OAArB,EAA8B,CAC/C,KAD+C,EACxC,uB;QACH,OAAO,KAAK,KAAK,CAAC,IAAD,EAAO,IAAP,EAAa,EAAE,MAAM,KAAK,CAAC,KAAD,CAA1B,C;OAF0B,CAA9B,C;;;IASzB,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,IAAK,KAAL,WAAxC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAAX,EAAuB,KAAvB,EAA8B,CAC/C,KAD+C,EACxC,0B;UACH,OAAO,EAAE,MAAM,KAAK,CAAC,IAAD,CAAM,IAAI,CAAC,QAAD,EAAW,IAAX,C;SAFa,CAA9B,C;;;IAU7B,IAAI,uBAAuB,gB;MACvB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,E;MACIB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,C;MAEIB,IAAI,CAAE,KAAL,CAAV,C;QACI,IAAI,CAAE,KAAL,CAAV,C;UAAa,OAAO,C;QAEpB,IAAI,KAAK,CAAE,GAAE,C;QACb,OAAO,EAA G,KAAL,CAAE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EAAF,GAAO,C;;MAG7C,OAAO,CAAE,KAAL,CAAE,GAAG,CAAE,KAAL,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;K;IAGzC,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAL,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAAX,EAAuB,MAAvB,EAA+B,CAChD,KADgD,EACzC,2B;UACH,OAAO,KAAK,UAAU,KAAK,KAAK,CAAC,IAAD,EAAO,eAAGB,IAAG,oBAA1B,C;SAFY,CAA/B,C;;;GAO/B,G;ECxXU;;;IAAZ,MAAM,KAAM,GAAE,CACV,KADU,EACH,OADG,EAEV,SAFU,EAEC,WAFD,EAGV,MAHU,EAGF,QAHE,C;EAMd,MAAM,WAAY,GAAE,2C;IACbB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;MACI,OAAO,kBAAkB,IAAI,KAAK,CAAC,UAAD,C;;IAGtC,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,OAAO,UAAU,CAAC,YAAD,C;;IAGrB,OAAO,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAAIC,EAA2C,YAA3C,C;G;EAG5B,MAAM,WAAY,GAAE,kD;IACbB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;MACI,kBAAkB,IAAI,KAAK,CAAC,UAAD,EAAa,KAAb,C;MAC3B,M;;IAGJ,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,UAAU,CAAC,YAAD,CAAe,GAAE,K;MAC3B,M;;IAGJ,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAAIC,EAA2C,YAA3C,EAAYD,KAAzD,C;G;EAGrB,iD;IACI,IAAI,IAAK,KAAL,KAAb,C;MAAoB,OAAO,I;IAE3B,IAAI,WAAW,IAAI,W;IACnB,IAAI,QAAS,IAAG,IAAhB,C;MACI,IAAI,aAAa,QAAQ,W;MACzB,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,UAAU,OAA9B,EAAuC,CAAC,EAAxC,C;QACI,IAAI,0BAA0B,CAAC,UAAU,CAAC,CAAD,CAAX,EAAGB,KAAhB,CAA9B,C;UACI,OAAO,I;;;IACnB,IAAI,iBAAiB,IAAI,UAAW,IAAG,IAAK,GAAE,MAAM,eAAe,CAAC,IAAI,UAAAL,CAAvB,GAA0C,I;IACtF,IAAI,mBAAmB,cAAe,IAAG,IAAK,GAAE,cAAc,YAAhB,GAA+B,I;IAC7E,OAAO,gBAAiB,IAAG,IAAK,IAAG,0BAA0B,CAAC,gBAAD,EAAMB,KAAAB,C;G;EASnD;;;IAAd,MAAM,OAAQ,GAAE,yB;IACZ,IAAI,KAAAM,KAAL,MAAd,C;MACI,QAAQ,OAAO,MAAf,C;aACS,Q;aACA,Q;aACA,S;aACA,U;UACD,OAAO,I;;UAE P,OAAO,MAAO,YAAW,M;;;IAIrC,IAAI,MAAO,IAAG,IAAK,IAAG,KAAM,IAAG,IAAK,KAAL,OAAO,MAAO,KAAI,QAAS,IAAG,OAAO,MAAO,KAAL,UAApD,CAApC,C;MACI,OAAO,K;;IAGX,IAAI,OAAO,KAAM,KAAL,UAAW,IAAG,MAAO,YAAW,KAArD,C;MACI,OAAO,I;;IAGX,IAAI,QAAQ,MAAM,eAAe,CAAC,KAAD,C;IACjC,IAAI,cAAc,KAAM,IAAG,IAAK,GAAE,KAAK,YAAP,GAAsB,I;IACtD,IAAI,WAAW,IAAG,IAAK,IAAG,YAAa,IAAG,WAA3C,C;MACI,IAAI,WAAW,WAAW,W;MAC1B,IAAI,QAAQ,KAAM,KAAL,MAAM,KAAK,O

AAjC,C;QACI,OAAO,MAAO,KAAI,K;;;IAI1B,IAAI,gBAAgB,KAAK,WAAzB;A,IAGA,IAAI,aAAc,IAAG,IAAr  
B,C;MACI,OAAO,MAAO,YAAW,K;;IAG7B,IAAI,aAAa,KAAM,KAAI,MAAM,KAAK,UAAW,IAAG,MAAM,  
YAAa,IAAG,IAA1E,C;MACI,OAAO,0BAA0B,CAAC,MAAM,YAAP,EAAqB,KAArB,C;;IAGrC,OAAO,K;G;EA  
GX,MAAM,SAAU,GAAE,a;IACd,OAAO,OAAO,CAAE,IAAG,QAAS,IAAG,CAAE,YAAW,MAAM,K;G;EAGt  
D,MAAM,OAAQ,GAAE,iB;IACZ,OAAO,KAAM,YAAW,MAAM,U;G;EAGIC,MAAM,aAAc,GAAE,iB;IACIB,I  
AAI,OAAO,OAAO,K;IAEIB,OAAO,IAAK,KAAI,QAAS,IACIB,IAAK,KAAI,SAAU,IACnB,MAAM,SAAS,CAA  
C,KAAD,CAAQ,IACvB,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM,OAAO,WAArB,C;G;EAGxB,MAAM,eA  
AgB,GAAE,iB;IACpB,OAAO,OAAO,KAAM,KAAI,QAAS,IAAG,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM  
,OAAO,aAArB,C;G;;;;;;aCnDV,gB;;;ICrE3C,gB;MAkBI,4B;MAjBA,aAA6C,E;MAC7C,gBAAGD,C;K;4EAG5  
C,Y;MAAQ,iB;K;+EAGR,Y;MAAQ,oB;K;qCAEZ,iB;MAAyC,OAAQ,0BAAR,YAAQ,EAAU,KAAM,QAAb,C;  
K;4BAEjD,iB;MAAmC,gBAAS,K;K;8BAE5C,Y;MAA+B,OAAG,MAAH,kBAA8B,IAA9B,C;K;8BAE/B,Y;MAA  
0B,gB;K;IAE1B,0B;MAAA,8B;K;;IAAA,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IDfJ,mC;MAC4C,oBAAa,MA  
AS,IAAT,CAAb,EAA6B,SAA7B,C;K;gEAE5C,yB;MAAA,mB;MAAA,6B;QAC2D,YAAa,QAAS,IAAT,C;QAIv  
D,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IALgF,IAKrE,  
CAAK,CAAL,C;;QALwC,OAOhD,K;O;KARX,C;gEAGA,uB;MAEiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,C  
AAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,KAAK,CAAL,C;;MAEf,OAAO,K;K;IAGX,kC;MAI  
iB,IAAN,I;MAFP,aAAsB,MAAe,IAAf,C;MAcTB,gBAAkB,c;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,E;  
QAAc,oBAAa,MAAb,EAAqB,KAArB,C;WAcD,WAFS,IAET,S;QAAS,a;;QAZA,U;QAAA,SAaqB,MAbf,OAAN,  
GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UAakC,MAZ9B,CAAM,CAAN,IAysC,IAZ3B,CAAK,CAAL,C;;QAYH,  
OAsB,M;;MAHIC,W;K;2EAOJ,yB;MAAA,iC;MAAA,6B;QACoF,YAAa,aAAa,IAAb,EAAMB,KAAmB,C;QAlB  
hF,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAiBoH,IAjBz  
G,CAAK,CAAL,C;;QAIbIE,OfzE,K;O;KAcX,C;IAGA,+B;MAKiB,IAAN,I;MAFP,aAAa,IAAO,WAAP,CAAm  
B,IAAnB,C;MACb,gBAAkB,W;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,YADS,IACT,EAAY,KAAZ,E;  
QAAqB,a;;QA1BZ,U;QAAA,SA2BkB,MA3BZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UA2B+B,MA1B  
3B,CAAM,CAAN,IA0BmC,IA1BxB,CAAK,CAAL,C;;QA0BH,OAAMB,M;;MAF/B,W;K;qEAMJ,yB;MAAA,2B;  
MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,UAAU,IAAV,EAAGB,IAAhB,C;QACC,OAAA,KAAM,OAAN,G  
AAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;UACT,KAAK,CAAC,CAAD,CAAR,GAAc,  
K;;QAEIB,OAAO,K;O;KARX,C;mFAWA,yB;MAAA,mB;MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,QAA  
Y,IAAZ,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;U  
ACT,KAAK,CAAC,CAAD,CAAR,GAAc,K;;QAEIB,OAAO,K;O;KARX,C;IAWA,+B;MAIiB,IAAN,I;MAFP,aA  
AsB,MAAY,IAAZ,C;MAcTB,gBAAkB,W;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,E;QAAc,oBAAa,MA  
Ab,K;WAcD,WAFS,IAET,S;QAAS,a;;QA3DA,U;QAAA,SA4DkB,MA5DZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,  
CAAV,mB;UA4D+B,MA3D3B,CAAM,CAAN,IA2DmC,IA3DxB,CAAK,CAAL,C;;QA2DH,OAAMB,M;;MAH/B  
,W;K;qEAOJ,yB;MAAA,2B;MAAA,6B;QAC2E,YAAa,UAAU,IAAV,EAAGB,KAAhB,C;QAjEvE,Q;QAAA,OA  
AA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAgEwG,IAhE7F,CAAK,CAA  
L,C;;QAgEwD,OA9DhE,K;O;KA6DX,C;IAGA,wC;MACiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MA  
Ab,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,S;;MAEf,OAAO,K;K;IEIFX,iC;MAAA,qC;MAEI,iBAC8B,Q;  
MAE9B,iBAC8B,sB;MAE9B,yBAEsC,MAAM,G;MAE5C,yBAEsC,CAAC,GAAD,GAAO,G;MAE7C,WAEwB,E  
AAE,MAAM,GAAR,C;MAExB,kBACuB,C;MAEvB,iBACsB,E,K;;IAxB1B,6C;MAAA,4C;QAAA,2B;;MAAA,q  
C;K;IA2BA,gC;MAAA,oC;MAEI,iBAC6B,O;MAE7B,iBAC6B,Y;MAE7B,yBAEqC,MAAO,G;MAE5C,yBAEqC,  
CAAC,GAAD,GAAQ,G;MAE7C,WAEuB,EAAE,MAAO,GAAT,C;MAEvB,kBACuB,C;MAEvB,iBACsB,E,K;;I  
AxB1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IA2BA,8B;MAAA,kC;MAEI,iBACqB,W;MAErB,iBACqB,U;M  
AErB,kBACuB,C;MAEvB,iBACsB,E,K;;IAZ1B,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;IAeA,+B;MAAA,mC;  
MAEI,iBACyB,MAAM,KAAT,U;MAEtB,iBACyB,MAAM,KAAT,U;MAEtB,kBACuB,C;MAEvB,iBACsB,E,K;;  
IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,gC;MAAA,oC;MAEI,iBACuB,U;MAEvB,iBACuB,K;MA  
EvB,kBACuB,C;MAEvB,iBACsB,E,K;;IAZ1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAeA,+B;MAAA,mC;M  
AEI,iBACsB,Q;MAEtB,iBACsB,G;MAEtB,kBACuB,C;MAEvB,iBACsB,C,K;;IAZ1B,2C;MAAA,0C;QAAA,yB;  
;MAAA,mC;K;IAeA,+B;MAAA,mC;MAEI,iBACmC,C;MAEnC,iBACmC,K;MAEnC,0BAC4C,K;MAE5C,0BAC



;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAO I,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,S AAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAA V,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;s GAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA ,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;8EAUA,gC;MAOW,sB;;QAkcS,Q;QA AhB,iD;UAAgB,cAAhB,e;UAA sB,IAIcH,SAkcO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9 C,qBAAO,I;;;MAncP,yB;K;gFAGJ,gC;MAOW,sB;;QAgcS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IAhcH,SAgc O,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAjcP,yB;K;gFAGJ,gC;MAOW,s B;;QA8bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IA9bH,SA8bO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;Y AAP,uB;;;QAC9C,qBAAO,I;;;MA/bP,yB;K;gFAGJ,gC;MAOW,sB;;QA4bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IA5bH,SA4bO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA7bP,yB;K;gFA GJ,gC;MAOW,sB;;QA0bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IA1bH,SA0bO,CAAU,OAAV,CAAJ,C;YAA wB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA3bP,yB;K;gFAGJ,gC;MAOW,sB;;QAwbS,Q;QAAhB,iD;UAA gB,cAAhB,e;UAA sB,IAxbH,SAwbO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I; ;MAzbP,yB;K;gFAGJ,gC;MAOW,sB;;QAsbS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IAtbH,SAsbO,CAAU,OA AV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAvbP,yB;K;gFAGJ,gC;MAOW,sB;;QAobS,Q ;QAAhB,iD;UAAgB,cAAhB,e;UAA sB,IApbH,SAobO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;Q AC9C,qBAAO,I;;;MARbP,yB;K;gFAGJ,yB;MAqbA,oC;MAAA,gC;MARbA,uC;QAOW,sB;;UAKbS,Q;UAAhB,iD; YAAgB,cAAhB,0B;YAA sB,IAIbH,SAkbO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qBA AO,I;;;QAnbP,yB;O;KAPJ,C;sFAUA,yB;MAi2CA,0D;MAAA,+C;MAj2CA,uC;QAOW,qB;;UA g2CO,Q;UAAA,O AAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAI2Cc,SA k2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAp2CP,wB;O;KAPJ,C;wFAU A,yB;MAo2CA,0D;MAAA,+C;MAp2CA,uC;QAOW,qB;;UAm2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;U AAd,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAr2Cc,SAq2CV,CAAU,OAAV,CAAJ,C;c AAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAv2CP,wB;O;KAPJ,C;wFAUA,yB;MAu2CA,0D;MAAA,+C; MAV2CA,uC;QAOW,qB;;UAs2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;Y ACV,cAAc,UAAK,KAAL,C;YACd,IAx2Cc,SAw2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UA E5B,oBAAO,I;;;QA12CP,wB;O;KAPJ,C;wFAUA,yB;MA02CA,0D;MAAA,+C;MA12CA,uC;QAOW,qB;;UAy2C O,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YA Cd,IA32Cc,SA22CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA72CP,wB;O; KAPJ,C;wFAUA,yB;MA62CA,0D;MAAA,+C;MA72CA,uC;QAOW,qB;;UA42CO,Q;UAAA,OAAa,SAAR,sBAA Q,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA92Cc,SA82CV,CAAU,OA AV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAh3CP,wB;O;KAPJ,C;wFAUA,yB;MAG3CA,0 D;MAAA,+C;MAh3CA,uC;QAOW,qB;;UA+2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAj3Cc,SAi3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;c AAP,sB;;;UAE5B,oBAAO,I;;;QAn3CP,wB;O;KAPJ,C;wFAUA,yB;MAM3CA,0D;MAAA,+C;MAN3CA,uC;QAO W,qB;;UAK3CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK ,KAAL,C;YACd,IAp3Cc,SAo3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA t3CP,wB;O;KAPJ,C;wFAUA,yB;MAS3CA,0D;MAAA,+C;MAT3CA,uC;QAOW,qB;;UAq3CO,Q;UAAA,OAAa,S AAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAv3Cc,SAu3CV, CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAz3CP,wB;O;KAPJ,C;wFAUA,yB; MAY3CA,0D;MAAA,+C;MAAA,oC;MAz3CA,uC;QAOW,qB;;UAw3CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb, W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA13Cc,SA03CV,CAAU,oBAAV,CAA J,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA53CP,wB;O;KAPJ,C;IAUA,0B;MAMI,IAovNO,qBAA Q,CApvNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,IAivNO,q BAAQ,CAjvNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,IA8u NO,qBAAQ,CA9uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,













W,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAA  
M,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC  
;QAMoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U  
ACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;Y  
ACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;Kaf  
X,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAmB,I;QACnB,YAA  
Y,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA  
AJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UA  
AY,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;  
MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,S  
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,  
SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,i  
E;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAqB,I;QACr  
B,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,I  
AAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KA  
AL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;M  
AAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACtB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,c  
AAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;  
YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,  
OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAu  
B,I;QACvB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,  
C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,C  
AAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4D;O;KafX,C;qFakBA,yB;MAAA,oC;MA  
AA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAy,K;QAC  
Z,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,K  
AAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;U  
AAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4E;O;KafX,C;IAkBA,iC;MAII,OAAW,qBAAQ,CAAZ,GAA  
e,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4  
B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAA  
W,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UA  
AK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;  
IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qB  
AAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,C  
AAL,CAAf,GAA4B,I;K;gGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAy,K;MACZ,wBAAgB,SAAhB,  
gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,  
SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;gGAGX,gC;MA  
MoB,Q;MAFhB,aAAoB,I;MACpB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI  
,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,  
CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAy  
,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAA  
J,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,  
OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAmB,I;MACnB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,  
cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;U  
ACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;M  
AFhB,aAAoB,I;MACpB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,O  
AAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,K  
AAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAy,K;MAC  
Z,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YA

AW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAA Y,OAAO,I;MACnB,OAAO ,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAsB,I;MActB,YAA Y,K;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QA AQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAA Y,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aA AuB,I;MACvB,YAA Y,K;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CA AJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;Q AAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I; QACpB,YAA Y,K;QACZ,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UACI,IAAI,UAAU,oBAAV, CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YACIB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C; UAAY,OAAO,I;QACnB,OAAO,M;O;KADx,C;IAiBA,4B;Me9qGI,IAAI,EfsrGI,KAAK,CetrGT,CAAJ,C;QACI,cf qrGc,sD;QeprGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfqrGV,OAAO,oBAAoB,gBAAV,mBAAO,CAAP,IAAU, EAAC,CAAd,CAApB,C;K;IAGX,8B;Me1rGI,IAAI,EfksGI,KAAK,CelsGT,CAAJ,C;QACI,cfisGc,sD;QehsGd,MA AM,gCAAYB,OAAQ,WAAjC,C;;MfisGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB, C;K;IAGX,8B;MetsGI,IAAI,Ef8sGI,KAAK,Ce9sGT,CAAJ,C;QACI,cf6sGc,sD;Qe5sGd,MAAM,gCAAYB,OAAQ, WAAjC,C;;Mf6sGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MeltG I,IAAI,Ef0tGI,KAAK,Ce1tGT,CAAJ,C;QACI,cfytGc,sD;QextGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfytGV,O AAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me9tGI,IAAI,EfsuGI,KAAK, CetuGT,CAAJ,C;QACI,cfquGc,sD;QepuGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfquGV,OAAO,sBAAoB,gBA AV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me1uGI,IAAI,EfkvGI,KAAK,CelvtGT,CAAJ,C;Q ACI,cfivGc,sD;QehvGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfivGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP, IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MetvGI,IAAI,Ef8vGI,KAAK,Ce9vGT,CAAJ,C;QACI,cf6vGc,sD;Qe5 vGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf6vGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAA d,CAApB,C;K;IAGX,8B;MelwGI,IAAI,Ef0wGI,KAAK,Ce1wGT,CAAJ,C;QACI,cfywGc,sD;QexwGd,MAAM,gC AAYB,OAAQ,WAAjC,C;;MfywGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;I AGX,8B;Me9wGI,IAAI,EfsxGI,KAAK,CetxGT,CAAJ,C;QACI,cfqxGc,sD;QepxGd,MAAM,gCAAYB,OAAQ,WA AjC,C;;MfqxGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,gC;Me1xGI,I AAI,EfkyGI,KAAK,CelyGT,CAAJ,C;QACI,cfiyGc,sD;QehyGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfiyGV,O AAO,gBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;MetyGI,IAAI,Ef8yGI,KAAK ,Ce9yGT,CAAJ,C;QACI,cf6yGc,sD;Qe5yGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf6yGV,OAAO,kBAAGB,gB AAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;MelzGI,IAAI,Ef0zGI,KAAK,Ce1zGT,CAAJ,C; QACI,cfyzGc,sD;QexzGd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfyzGV,OAAO,kBAAGB,gBAAV,mBAAO,CA AP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me9zGI,IAAI,Efs0GI,KAAK,Ce0GT,CAAJ,C;QACI,cfq0Gc,sD; Qep0Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfq0GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,C AAd,CAAhB,C;K;IAGX,kC;Me10GI,IAAI,Efk1GI,KAAK,Ce11GT,CAAJ,C;QACI,cfi1Gc,sD;Qeh1Gd,MAAM,gC AAYB,OAAQ,WAAjC,C;;Mfi1GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;I AGX,kC;Met1GI,IAAI,Ef81GI,KAAK,Ce91GT,CAAJ,C;QACI,cf61Gc,sD;Qe51Gd,MAAM,gCAAYB,OAAQ,WA AjC,C;;Mf61GV,OAAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mel2GI,I AAI,Ef02GI,KAAK,Ce12GT,CAAJ,C;QACI,cfy2Gc,sD;Qex2Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfy2GV,O AAO,kBAAGB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me92GI,IAAI,Efs3GI,KAA K,Ce03GT,CAAJ,C;QACI,cfq3Gc,sD;Qep3Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfq3GV,OAAO,kBAAGB,gB AAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me13GI,IAAI,Efk4GI,KAAK,Ce14GT,CAAJ,C; QACI,cfi4Gc,sD;Qeh4Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfi4GV,OAAO,kBAAGB,gBAAV,mBAAO,CAA P,IAAU,EAAC,CAAd,CAAhB,C;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wB AAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAA Q,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,qD;MAAA,uC;QAM I,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,g BAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MA A,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YA



BAAS,gB;QAwgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAxgBc,SAwgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAl,OAAJ,C;;QAxgB1D,OAYgBO,W;O;KA/gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAYgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAzgBe,SAygBX,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAl,OAAJ,C;;QAZgB1D,OA0gBO,W;O;KAhhBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA0gBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IA1gBgB,SA0gBZ,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAl,OAAJ,C;;QA1gB1D,OA2gBO,W;O;KAjhBX,C;oFASA,yB;MAAA,+D;MA2gBA,oC;MAAA,gC;MA3gBA,uC;QAMW,kBAAS,gB;QA2gBA,Q;QAAhB,iD;UAAgB,cAAhB,0B;UAAsB,IA3gBa,SA2gBT,CAAU,oBAAV,CAAJ,C;YAAwB,WAAY,WAAl,oBAAJ,C;;QA3gB1D,OA4gBO,W;O;KAlhBX,C;gGASA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QA0iTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAAt8SI,IApGmC,SAoG/B,EAs8SkB,cAt8SIB,EAs8SkB,sBAAt8SIB,WAs8S2B,IAAt8S3B,CAAJ,C;YAA2C,sBAAs8SZ,IAAt8SY,C;;QApG/C,OAsGO,W;O;KA9GX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAYiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA18SI,IAvGsC,SAuGIC,EAk8SkB,cAl8SIB,EAk8SkB,sBA18SIB,WAk8S2B,IA18S3B,CAAJ,C;YAA2C,sBAk8SZ,IA18SY,C;;QAvG/C,OAYGO,W;O;KAjHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAWiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA97SI,IA1GuC,SA0GnC,EA87SkB,cA97SIB,EA87SkB,sBA97SIB,WA87S2B,IA97S3B,CAAJ,C;YAA2C,sBA87SZ,IA97SY,C;;QA1G/C,OA4GO,W;O;KApHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAuiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA17SI,IA7GqC,SA6GjC,EA07SkB,cA17SIB,EA07SkB,sBA17SIB,WA07S2B,IA17S3B,CAAJ,C;YAA2C,sBA07SZ,IA17SY,C;;QA7G/C,OA+GO,W;O;KAvHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAsiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAAt7SI,IAhHsC,SAgHIC,EAs7SkB,cAt7SIB,EAs7SkB,sBAAt7SIB,WAs7S2B,IAAt7S3B,CAAJ,C;YAA2C,sBAAs7SZ,IAAt7SY,C;;QAhH/C,OAKHO,W;O;KA1HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAqiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA17SI,IAAnHuC,SAmHnC,EAk7SkB,cAl7SIB,EAk7SkB,sBA17SIB,WAk7S2B,IA17S3B,CAAJ,C;YAA2C,sBAk7SZ,IA17SY,C;;QAnH/C,OAqHO,W;O;KA7HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAoiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA96SI,IAthwC,SAsHpC,EA86SkB,cA96SIB,EA86SkB,sBA96SIB,WAs8S2B,IA96S3B,CAAJ,C;YAA2C,sBA86SZ,IA96SY,C;;QAtH/C,OAwhO,W;O;KAhIX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAmiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA16SI,IAzHyC,SAyHrC,EA06SkB,cA16SIB,EA06SkB,sBA16SIB,WA06S2B,IA16S3B,CAAJ,C;YAA2C,sBA06SZ,IA16SY,C;;QAZh/C,OA2HO,W;O;KAnIX,C;kGAWA,yB;MAAA,+D;MA2HA,gC;MAw6SA,oC;MAniTA,uC;QAQW,kBAAgB,gB;QAkiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAAt6S/B,IA5HsC,SA4HIC,CAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,sBAAI,OAAJ,C;;QA5H/C,OAs8HO,W;O;KAtIX,C;oGAWA,6C;MA+8SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAt8SI,IAAI,WAs8SkB,cAt8SIB,EAs8SkB,sBAAt8SIB,WAs8S2B,IAAt8S3B,CAAJ,C;UAA2C,sBAAs8SZ,IAAt8SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MA28SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA18SI,IAAI,WAk8SkB,cAl8SIB,EAk8SkB,sBA18SIB,WAk8S2B,IA18S3B,CAAJ,C;UAA2C,sBAk8SZ,IA18SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAu8SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA97SI,IAAI,WA87SkB,cA97SIB,EA87SkB,sBA97SIB,WA87S2B,IA97S3B,CAAJ,C;UAA2C,sBA87SZ,IA97SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MAm8SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA17SI,IAAI,WA07SkB,cA17SIB,EA07SkB,sBA17SIB,WA07S2B,IA17S3B,CAAJ,C;UAA2C,sBA07SZ,IA17SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA+7SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAt7SI,IAAI,WAs7SkB,cAt7SIB,EAs7SkB,sBAAt7SIB,WAs7S2B,IAAt7S3B,CAAJ,C;UAA2C,sBAAs7SZ,IAAt7SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA27SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA17SI,IAAI,WAk7SkB,cAl7SIB,EAk7SkB,sBA17SIB,WAk7S2B,IA17S3B,CAAJ,C;UAA2C,sBAk7SZ,IA17SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAu7SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA96SI,IAAI,WAs8S2B,cA96SIB,EA86SkB,sBA96SIB,WA86S2B,IA96S3B,CAAJ,C;UAA2C,sBA86SZ,IA96SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAm7SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA16SI,IAAI,WA06SkB,cA16SIB,EA06SkB,sBA16SIB,WA06S2B,IA16S3B,CAAJ,C;UAA2C,sBA06SZ,IA16SY,C;;MAE/C,OAAO,W;K;sGAGX,yB;MAAA,gC;MAw6SA,oC;MAx6SA,oD;QA+6SiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAAt6S/B,IAAI,UAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,sBAAI,OAAJ,C;;QAE/C,OAAO,W;O;KAXX,C;sGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,Q;Q



AAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,YAAJ,C;YAAkB,WAAY,WAAl,OAAJ,C;;QATpD,OAuO,W;O;KAhB X,C;OGASA,4C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,YAAJ,C;UA AkB,WAAY,WAAl,OAAJ,C;;MACpD,OAAO,W;K;wFAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA oGH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CAPGS,SAoGR,CAAU,OAAV,CAAL,C;YAAyB,WAAY,W AAI,OAAJ,C;;QApG3D,OAqGO,W;O;KA3GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAqG H,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CARGY,SAqGX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAA I,OAAJ,C;;QArG3D,OAsGO,W;O;KA5GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAsGH,Q; QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CAtGa,SAsGZ,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OA AJ,C;;QATG3D,OAuGO,W;O;KA7GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAUgh,Q;QAA hB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CavGW,SAuGV,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OAAJ, C;;QAvG3D,OAwGO,W;O;KA9GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAwGH,Q;QAAh B,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CaxGY,SAwGX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OAAJ,C; ;QAxG3D,OAYGO,W;O;KA/GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAYGH,Q;QAAhB,iD ;UAAgB,cAAhB,e;UAAsB,IAAI,CAzGa,SAYGZ,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OAAJ,C;;QAZ G3D,OA0GO,W;O;KAhHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA0GH,Q;QAAhB,iD;UA AgB,cAAhB,e;UAAsB,IAAI,CA1Gc,SA0Gb,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OAAJ,C;;QA1G3D, OA2GO,W;O;KAjHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA2GH,Q;QAAhB,iD;UAAgB,c AAhB,e;UAAsB,IAAI,CA3Ge,SA2Gd,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAl,OAAJ,C;;QA3G3D,OA4 GO,W;O;KAIHX,C;0FASA,yB;MAAA,+D;MA4GA,oC;MAAA,gC;MA5GA,uC;QAMW,kBAAY,gB;QA4GH,Q; QAAhB,iD;UAAgB,cAAhB,0B;UAAsB,IAAI,CA5GY,SA4GX,CAAU,oBAAV,CAAL,C;YAAyB,WAAY,WAAl, oBAAJ,C;;QA5G3D,OA6GO,W;O;KANHX,C;IASA,kC;MAMI,OAAO,2BAAGB,gBAhB,C;K;IAGX,iD;MAMo B,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,eAAJ,C;UAAqB,WAAY,WAAl,OA AJ,C;;MACvD,OAAO,W;K;4FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q AAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C; MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL ,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB ;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC 3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI, CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q; MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB, WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,c AAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO ,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UA AU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wB AAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WA Al,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAhB,wBAAGB, SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,WAAY, WAAl,oBAAJ,C;;QAC3D,OAAO,W;O;KAPX,C;sFAUA,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB, cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1D,OAAO,W;K; wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,C AAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAA hB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1 D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI, UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAh B,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl ,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB, M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;M AMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAw

B, WAAY, WAAI, OAAJ, C; ; MAC1D, OAAO, W; K; wFAGX, 6C; MAMoB, Q; MAAhB, wBAAgB, SAAhB, gB; QAAgB, cAAA, SAAhB, M; QAAsB, IAAI, UAAU, OAAV, CAAJ, C; UAAwB, WAAY, WAAI, OAAJ, C; ; MAC1D, OAAO, W; K; wFAGX, yB; MAAA, oC; MAAA, gC; MAAA, oD; QAMoB, Q; QAAhB, wBAAgB, SAAhB, gB; UAAgB, cAAhB, UAAgB, SAAhB, O; UAAsB, IAAI, UAAU, oBAAV, CAAJ, C; YAAwB, WAAY, WAAI, oBAAJ, C; ; QAC1D, OAAO, W; O; KAPX, C; IAUA, mC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OKvtIe, W; ; MLwtItC, OAA4D, OAArD, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CAAqD, C; K; IAGhE, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OK/Ie, W; ; MLguItC, OgBpsIsC, OhBosI/B, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsI+B, C; K; IhBusI1C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OKvuIe, W; ; MLwuItC, OgBpsIuC, OhBosIhC, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsIgC, C; K; IhBusI3C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OK/uIe, W; ; MLgvItC, OgBpsIqC, OhBosI9B, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsI8B, C; K; IhBusIzC, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OKvvIe, W; ; MLwvItC, OgBpsIsC, OhBosI/B, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsI+B, C; K; IhBusI1C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OK/vIe, W; ; MLgwItC, OgBpsIuC, OhBosIhC, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsIgC, C; K; IhBusI3C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OKvwIe, W; ; MLwwItC, OgBpsIwC, OhBosIjC, yBAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsIiC, C; K; IhBusI5C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OK/wIe, W; ; MLgxItC, OgBpsIyC, OhBosIiC, 0BAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CgBpsIkC, C; K; IhBusI7C, qC; MAII, IAAI, OAAQ, UAAZ, C; QAAuB, OKvxIe, W; ; MLwxItC, OAA4D, SAArD, 0BAAY, OAAQ, MAApB, EAA2B, OAAQ, aAAR, GAAuB, CAAvB, IAA3B, CAAqD, C; K; IAGhE, qC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAaA, IAAb, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, qC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAgB, IAAhB, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, sC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAiB, IAAjB, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, sC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAb, IA Af, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, sC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAgB, IAAhB, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, sC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAmB, IAAnB, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, UAAI, KAAJ, CAAJ, C; ; MAET, OAAO, I; K; IAGX, sC; MAOkB, Q; MAHd, WAAmB, wBAAR, OAAQ, EAAwB, EAAxB, C; MACnB, IAAI, SAAQ, CAAZ, C; QAAe, OAAO, W; MACtB, WAAW, iBAAgB, IAAhB, C; MACG, yB; MAAd, OAAc, cAAAd, C; QAAc, uB; QACV, IAAK, WAAI, sBAAI, KAAJ, EAAJ, C; ; MAET, OAAO, I; K; IAGX, wC; MAMwB, UACT, M; MAHX, aAAa, aAAa, SAAb, EAAmB, OAAQ, KAA3B, C; MACb, kBAakB, C; MACE, yB; MAApB, OAAoB, cAApB, C; QAAoB, 6B; QChB, OAAO, oBAAP, EAAO, 4BAAP, YAAwB, UAAK, WAAL, C; ; MAE5B, OAAO, M; K; IAGX, 0C; MAMwB, UACT, M; MAHX, aAAa, cAAU, OAAQ, KAAIB, C; MACb, kBAakB, C; MACE, yB; MAApB, OAAoB, cAApB, C; QAAoB, 6B; QChB, OAAO, oBAAP, EAAO, 4BAAP, YAAwB, UAAK, WAAL, C; ; MAE5B, OAAO, M; K; IAGX, 0C; MAMwB, UACT, M; MAHX, aAAa, eAAW, OAAQ, KAAjB, C; MACb, kBAakB, C; MACE, yB; MAApB, OAAoB, cAApB, C; QAAoB, 6B; QChB, OAAO, oBAAP, EAAO, 4BAAP, YAAwB, UAAK, WAAL, C; ; MAE5B, OAAO, M; K; IAGX, 0C; MAMwB, UACT, M; MAHX, aAAa, eAAS, OAAQ, KAAjB, C; MACb, kBAakB, C; MACE, yB; MAApB, OAAoB, cAApB, C; QAAoB, 6B; QChB, OAAO, oBAAP, EAAO, 4BAAP, YAAwB, UAAK, WAAL

,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAAnB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAAnB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,yBAAY,CAAZ,EAAe,CAAF,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,cAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,eAAS,CAAT,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,4B;MAciB,Q;MeloJb,IAAI,Ef4nJI,KAAK,Ce5nJT,CAAJ,C;QACI,cf2nJc,sD;Qe1nJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf2nJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,iB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAa,CAAb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MexpJb,IAAI,EfkpJI,KAAK,CelpJT,CAAJ,C;QACI,cfipJc,sD;QehpJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfipJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me9qJb,IAAI,EfwqJI,KAAK,CexqJT,CAAJ,C;QACI,cfuqJc,sD;QetqJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfuqJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me1tJb,IAAI,EfotJI,KAAK,CeptJT,CAAJ,C;QACI,cfmtJc,sD;QeltJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfmtJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MehvJb,IAAI,Ef0uJI,KAAK,Ce1uJT,CAAJ,C;QACI,cfyuJc,sD;QexuJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfyuJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gB

AAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAaA,SAAb,gB;QAaA,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MetwJb,IAAI,EfgwJI,KAAK,CehwJT,CAAJ,C;QACI,cf+vJc,sD;Qe9vJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf+vJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAaKb,CAAlB,C;MACX,wBAaA,SAAb,gB;QAaA,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me5xJb,IAAI,EfsxJI,KAAK,CetxJT,CAAJ,C;QACI,cfqxJc,sD;QepxJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfqxJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAmB,CAAnB,C;MACX,wBAaA,SAAb,gB;QAaA,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me1zJl,IAAI,Efk0JI,KAAK,Cel0JT,CAAJ,C;QACI,cfi0Jc,sD;Qeh0Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfi0JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,iB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAaA,CAAb,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me70JI,IAAI,Efq1JI,KAAK,Cer1JT,CAAJ,C;QACI,cf01Jc,sD;Qen1Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfo1JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Meh2JI,IAAI,Efw2JI,KAAK,Cex2JT,CAAJ,C;QACI,cfu2Jc,sD;Qet2Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfu2JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA Ae,CAAf,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Met4JI,IAAI,Ef84JI,KAAK,Ce94JT,CAAJ,C;QACI,cf64Jc,sD;Qe54Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf64JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mez5JI,IAAI,Ef6JI,KAAK,Cej6JT,CAAJ,C;QACI,cf6Jc,sD;Qe/5Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfg6JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AiB,CAAjB,C;MACX,iBA Ac,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me56JI,IAAI,Efo7JI,KAAK,Cep7JT,CAAJ,C;QACI,cfm7Jc,sD;Qel7Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfm7JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAaKb,CAAlB,C;MACX,iBA Ac,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAL,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me/7JI,IAAI,Efu8JI,KAAK,Cev8JT,CAAJ,C;QACI,cf8Jc,sD;Qer8Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfs8JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MAC





B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAA,Y,OOAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;  
QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,  
KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;K;IAIR,6B;  
MAII,IA+nEO,qBAAQ,CA/nEf,C;QAAe,OAAO,W;MACtB,WAAW,wB;MACN,WAAL,IAAK,C;MACL,OAAO,  
I;K;IAGX,+B;MAII,IA6nEO,qBAAQ,CA7nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;  
MACL,OAAO,I;K;IAGX,+B;MAII,IA2nEO,qBAAQ,CA3nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,W  
AAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAynEO,qBAAQ,CAznEf,C;QAAe,OAAO,W;MACtB,WAAW,  
0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAunEO,qBAAQ,CAvnEf,C;QAAe,OAAO,W;M  
ACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAqnEO,qBAAQ,CArnEf,C;QAAe  
,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAmnEO,qBAAQ,C  
AnnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAin  
EO,qBAAQ,CAjnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+  
B;MAII,IA+mEO,qBAAQ,CA/mEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OA  
AO,I;K;IAGX,kC;MAII,IAqiEO,qBAAQ,CAriEf,C;QAAe,OAAO,S;MACtB,aAAa,aAAa,SAAb,EAAmB,gBAAn  
B,C;MACb,gBAAgB,wB;MACHb,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UA  
AK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAiiEO,qBAAQ,CAjiEf,C;QAAe,OAAO,S;MACtB,aAAa,cA  
AU,gBAAV,C;MACb,gBAAgB,0B;MACHb,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IA  
AwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6hEO,qBAAQ,CA7hEf,C;QAAe,OAAO,S;MACt  
B,aAAa,eAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CA  
AZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAyhEO,qBAAQ,CAzhEf,C;QAAe,OA  
AO,S;MACtB,aAAa,eAAS,gBAAT,C;MACb,gBAAgB,0B;MACHb,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,  
YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqhEO,qBAAQ,CArhEf,C;  
QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,AAAU,CAAV,OAAa,SAAb,M;Q  
ACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAihEO,qBAAQ  
,CAjhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,AAAU,CAAV,OAAa,  
SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6g  
EO,qBAAQ,CA7gEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAA,Y,gBAAZ,C;MACb,gBAAgB,0B;MACHb,AAAU,C  
AAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,o  
C;MAII,IAygEO,qBAAQ,CAzgEf,C;QAAe,OAAO,S;MACtB,aAAa,oBAAa,gBAAb,C;MACb,gBAAgB,0B;MAC  
hB,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,  
M;K;IAGX,oC;MAII,IAqgEO,qBAAQ,CArgEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBA  
AgB,0B;MACHb,AAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAA,Y,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MA  
C5B,OAAO,M;K;IAGX,4B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,s  
BAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,  
sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAK  
I,sBAAQ,4BAAR,C;K;IAGJ,sC;MAOI,AAAU,wBAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAA  
R,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAA  
L,IAAU,I;K;IAIIB,sC;MAOI,AAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAA  
R,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;  
K;IAIIB,uC;MAOI,AAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf  
,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;K;IAIIB,u  
C;MAOI,AAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,  
UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;K;IAIIB,uC;MAOI  
,AAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,C  
AAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;K;IAIIB,uC;MAOI,AAAU,0  
BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;Q  
ACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;K;IAIIB,uC;MAOI,AAAU,0BAAV,O  
AA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UA

AK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CA  
A3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAA  
L,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;  
QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,  
UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;kFAIIB,yB;MAAA,oD;MiB15LA,sC;MAAA,oC;MAAA,uBA  
Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA  
2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBm5Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAAc,oBiBz5Ld,eA  
AW,iBjBy5LsB,QiBz5LtB,CAAX,CjBy5Lc,C;;O;KANIB,C;sGASA,yB;MAAA,oD;MiBh5LA,sC;MAAA,oC;MA  
AA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAA  
d,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBy4Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAAc,oBiB/4L  
d,eAAW,2BjB+4LgC,QiB/4LhC,CAAX,CjB+4Lc,C;;O;KANIB,C;IASA,mC;MAMI,oBAAS,cAAT,C;K;IAGJ,qC;  
MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;  
K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,i  
B;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CA  
AX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,2B;MAMI,OAA  
qB,OAAAd,sBAAc,C;K;IAGzB,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBnhMhC,WlBmhMgC,C;  
K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkB1hMhC,WlB0hMgC,C;K;IAG3C,6B;MAI0B,  
kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBjiMhC,WlBiiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB  
;MAA9B,OAAuC,OkBxiMhC,WlBwiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,Ok  
B/iMhC,WlB+iMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBtjMhC,WlBsjMgC,C;  
K;IAG3C,6B;MAI0B,kBAAf,0B;MAAuB,mB;MAA9B,OAAuC,OkB7jMhC,WlB6jMgC,C;K;IAG3C,gC;MAMI,I  
A6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3jKiB,Q;MhB2jKK,mB;MAA7B,OkBvkMO,W;  
K;IIB0kMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBzjKiB,Q;MhByjKK,iB;M  
AA7B,OkB/kMO,W;K;IIBklMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBvjKi  
B,Q;MhBujKK,iB;MAA7B,OkBvlMO,W;K;IIB0lMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MA  
CD,kBAAd,SgBrjKiB,Q;MhBqjKK,iB;MAA7B,OkB/lMO,W;K;IIBkmMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;  
QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,mB;MAA7B,OkBvmMO,W;K;IIB0mMX,kC;MAII,IA6  
kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBljKiB,Q;MhBkjKK,iB;MAA7B,OkB/mMO,W;K;I  
IBknMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBhjKiB,Q;MhBgjKK,iB;MAA  
7B,OkBvnMO,W;K;IIB0nMX,kC;MAII,IAqlDO,qBAAQ,CARlDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SA  
AK,C;MAAiB,iB;MAA7B,OkB/nMO,W;K;IIBkoMX,0C;MAMI,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MA  
CD,kBAAd,SgB7nKiB,Q;MhB6nKK,sBAAS,cAAT,C;MAA7B,OkBzoMO,W;K;IIB4oMX,4C;MAII,IA2gDO,qB  
AAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3nKiB,Q;MhB2nKK,6B;MAA7B,OkBjpMO,W;K;IIBopMX  
,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBznKiB,Q;MhBynKK,6B;MAA7B,Ok  
BzpMO,W;K;IIB4pMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBvnKiB,Q;Mh  
BunKK,6B;MAA7B,OkBjqMO,W;K;IIBoqMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kB  
AAT,UAAL,SAAK,C;MAAiB,6B;MAA7B,OkBzqMO,W;K;IIB4qMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA  
Ae,OAAO,S;MACD,kBAAd,SgBpnKiB,Q;MhBonKK,6B;MAA7B,OkBjrMO,W;K;IIBorMX,4C;MAII,IA2gDO,q  
BAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBlNKiB,Q;MhBknKK,6B;MAA7B,OkBzrMO,W;K;IIB4rM  
X,4C;MAII,IAmhDO,qBAAQ,CAnhDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,6B;MAA7B,  
OkBjsMO,W;K;IIBosMX,gD;MAMI,IAy8CO,qBAAQ,CAz8Cf,C;QAAe,OAAO,S;MACD,kBAAd,SgB/rKiB,Q;M  
hB+rKK,iC;MAA7B,OkB3sMO,W;K;sFIB8sMX,yB;MAAA,wD;MiBnsMA,sC;MAAA,oC;MAAA,uBAOe,yB;Q  
ArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4D  
M,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB4rMf,sC;QAQI,OAAO,sBiB7sMP,eAAW,iBjBosMiB,QiBpsMjB,CAA  
X,CjBosMO,C;O;KARX,C;wFAWA,yB;MAAA,wD;MiB9sMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAq  
Ee,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,C  
AA3B,C;W;S;OA4DI,C;MjBusMf,sC;QAMI,OAAO,sBiB7sMP,eAAW,iBjB6sMiB,QiB7sMjB,CAAX,CjB6sMO,  
C;O;KANX,C;wFASA,yB;MAAA,wD;MiBvtMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,



uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBgtMf,sC;QAMI,OAAO,sBiBttMP,eAAW,iBjBstMiB,QiBttMjB,CAAX,CjBstMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBhuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBytMf,sC;QAMI,OAAO,sBiB/tMP,eAAW,iBjB+tMiB,QiB/tMjB,CAAX,CjB+tMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBzuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBkuMf,sC;QAMI,OAAO,sBiBxuMP,eAAW,iBjBwuMiB,QiBxuMjB,CAAX,CjBwuMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBlvMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB2uMf,sC;QAMI,OAAO,sBiBjvMP,eAAW,iBjBivMiB,QiBjvMjB,CAAX,CjBivMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB3vMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBovMf,sC;QAMI,OAAO,sBiB1vMP,eAAW,iBjB0vMiB,QiB1vMjB,CAAX,CjB0vMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBpwMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB6vMf,sC;QAMI,OAAO,sBiBnwMP,eAAW,iBjBmwMiB,QiBnwMjB,CAAX,CjBmwMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB7wMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBswMf,sC;QAMI,OAAO,sBiB5wMP,eAAW,iBjB4wMiB,QiB5wMjB,CAAX,CjB4wMO,C;O;KANX,C;0GASA,yB;MAAA,wD;MiBnwMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB4vMf,sC;QAMI,OAAO,sBiB1wMP,eAAW,2BjBkwM2B,QiB1wM3B,CAAX,CjBkwMO,C;O;KANX,C;4GASA,yB;MAAA,wD;MiB5wMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBqwmf,sC;QAI,OAAO,sBiBzwMP,eAAW,2BjBywM2B,QiBzwM3B,CAAX,CjBywMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBnxMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB4wMf,sC;QAI,OAAO,sBiBhxMP,eAAW,2BjBgxM2B,QiBhxM3B,CAAX,CjBgxMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB1xMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBmxMf,sC;QAI,OAAO,sBiBvxMP,eAAW,2BjBuxM2B,QiBvxM3B,CAAX,CjBuxMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBjyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB0xmMf,sC;QAI,OAAO,sBiB9xMP,eAAW,2BjB8xM2B,QiB9xM3B,CAAX,CjB8xMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBxyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB4wMf,sC;QAI,OAAO,sBiB5yMP,eAAW,2BjB4yM2B,QiB5yM3B,CAAX,CjB4yMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBtzMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB+yMf,sC;QAI,OAAO,sBiBnzMP,eAAW,2BjBmzM2B,QiBnzM3B,CAAX,CjBmzMMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB7zMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBszMf,sC;QAI,OAAO,sBiB1zMP,eAAW,2BjB0zM2B,QiB1zM3B,CAAX,CjB0zMMO,C;O;KAJX,C;IAOA,qC;MAMI,OAAO,sBAAW,cAAx,C;K;IAGX,uC;MAIoB,kBgBz1KQ,iB;MhBy1KA,iB;MAAxB,OAAiC,WkBt3M1B,WIBs3M0B,C;K;IAGrC,uC;MAIoB,kBgBt1KQ,iB;MhBs1KA,iB;MAAxB,OAAiC,WkBt3M1B,WIBs3M0B,C;K;IAGrC,uC;MAIoB,kBgBn1KQ,iB;MhBm1KA,iB;MAAxB,OAAiC,WkB73M1B,WIB63M

0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB;MAAiB,mB;MAAxB,OAAiC,WkBp4M1B,WIBo4M0B,C;K;IAGrC,uC;M  
AIoB,kBgB90KQ,iB;MhB80KA,iB;MAAxB,OAAiC,WkB34M1B,WIB24M0B,C;K;IAGrC,uC;MAIoB,kBgB30K  
Q,iB;MhB20KA,iB;MAAxB,OAAiC,WkB15M1B,WIBk5M0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB;MAAiB,iB;MA  
AxB,OAAiC,WkBz5M1B,WIBy5M0B,C;K;IAGrC,2C;MAMI,OAAmC,OAA5B,2BAAGB,UAAhB,CAA4B,C;K;I  
AGvC,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBz6M9C,WIBy6M8C,C;K;IAGzD,6C;MAI0B,kB  
AAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBh7M9C,WIBg7M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;M  
AA9B,OAAqD,OkBv7M9C,WIBu7M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkB97  
M9C,WIB87M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBr8M9C,WIBq8M8C,C;K;  
IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkB58M9C,WIB48M8C,C;K;IAGzD,6C;MAI0B,kB  
AAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBn9M9C,WIBm9M8C,C;K;IAGzD,6C;MAI0B,kBAAf,0B;MAAuB,iC;  
MAA9B,OAAqD,OkB19M9C,WIB09M8C,C;K;IAkoCrD,gC;MAAQ,oBAAS,CAAT,EAAY,wBAAZ,C;K;IAMR,  
kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;  
MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MA  
AQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,  
oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAFAEZ,qB;MAKI,OA  
AO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;  
MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sF  
AGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,  
C;K;0FAGnB,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CA  
wER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,C  
AwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,  
CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,  
CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;IAOP,kC;MAAQ,0BAAO,CAAP,I;K;IA  
MR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;  
IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;  
IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IA8TZ,yD;MAcI,sBAAS,cAAT,EA  
AyB,SAAZB,EAAoC,OAAPC,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,qBAAQ,SAAR,EA  
AAmB,OAAnB,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAnB,C;K;IAGJ,0  
D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAnB,C;K;IAGJ,0D;MAYI,mBA  
AK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EA  
AGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,  
C;MACA,sBAAQ,SAAR,EAAMB,OAAnB,C;K;IA2B0B,oD;MAAA,wB;QAAW,2BAAK,KAAL,C;O;K;IAJzC,m  
C;MAII,OAAO,qBAAa,gBAAb,EAAMB,gCAAnB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;I  
AJtC,gC;MAII,OAAO,+BAAU,gBAAV,GAAGB,6BAAhB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C  
;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOkB,kD;MAAA,wB;QAAW,0BAAK,KA  
AL,C;O;K;IAJxkC,kC;MAII,OAAO,kCAAY,gBAAZ,GAakB,+BAAIB,C;K;IAOiB,gD;MAAA,wB;QAAW,yBAA  
K,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,kCAAW,gBAAX,GAAiB,8BAAjB,C;K;IAOe,4C;MAAA,wB;QAAW,uB  
AAK,KAAL,C;O;K;IAJrC,+B;MAII,OAAO,gCAAS,gBAAT,GAAe,4BAAf,C;K;IAOgB,8C;MAAA,wB;QAAW,  
wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOiB,gD;MAAA,wB;Q  
AAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,gCAAW,gBAAX,GAAiB,8BAAjB,C;K;wFA2CX,yB;MAA  
A,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,k  
BAAY,mBAAoB,QAAPB,C;QAYqBH,Q;QAAB,iD;UAAGB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAA  
V,C;UM3+QnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAAR,B,C;QNi0PA,OA4qBO,W;O;KAXrBX,C;0FAeA,yB;  
MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QA  
C1B,kBAAY,mBAAoB,QAAPB,C;QAYqBH,Q;QAAB,iD;UAAGB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,  
OAAV,C;UM1/QnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAAR,B,C;QNg1PA,OA4qBO,W;O;KAXrBX,C;0FAe  
A,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,

C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,C  
AAU,OAAV,C;UMzgRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN+1PA,OA4qBO,W;O;KAXrBX,C;  
0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,E  
AAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0q  
B/B,CAAU,OAAV,C;UMxhRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN82PA,OA4qBO,W;O;KAXr  
BX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,E  
AAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C  
,SA0qB/B,CAAU,OAAV,C;UMviRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN63PA,OA4qBO,W;O;  
KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAA  
kB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1q  
B8C,SA0qB/B,CAAU,OAAV,C;UMtjRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN44PA,OA4qBO,  
W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,  
CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,  
WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMrkRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN25PA,OA4  
qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gB  
AAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;U  
ACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMplRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN06PA,  
OA4qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,gC;MA5qBA,uC;  
QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q  
;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WA1qB8C,SA0qB/B,CAAU,oBAAV,C;UMnmRnB,wBAAI,IAAK,MAAT  
,EAAGB,IAAK,OAArB,C;;QNY7PA,OA4qBO,W;O;KAXrBX,C;4FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M  
AAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA  
mQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aApQoC,WAoQhC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C  
;;QApQhB,OAsQO,W;O;KAlRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAI  
B,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc,mBAAuB,QAAvB,C;QAoQL,Q;QAAhB,iD;UAAgB,c  
AAhB,e;UACI,WAAY,aArQuC,WAqQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QArQhB,OAuQO,W;O;KAn  
RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,E  
AAc,EAAd,C;QAC1B,kBAAc,mBAAwB,QAAxB,C;QAqQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAt  
QwC,WAsQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAtQhB,OAwQO,W;O;KApRX,C;8FAeA,yB;MAAA,0  
D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBA  
Ac,mBAAAsB,QAAtB,C;QAsQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAvQsC,WAuQIC,CAAY,OAAZ,  
CAAJ,EAA0B,OAA1B,C;;QAvQhB,OAYQO,W;O;KArRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAA  
A,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc,mBAAuB,QAAvB,C;QAuQ  
L,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAxQuC,WAwQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;Q  
AxQhB,OA0QO,W;O;KAtRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,  
YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc,mBAAwB,QAAxB,C;QAwQL,Q;QAAhB,iD;UAAgB,c  
AAhB,e;UACI,WAAY,aAzQwC,WAYqPc,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAzQhB,OA2QO,W;O;KA  
vRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,E  
AAc,EAAd,C;QAC1B,kBAAc,mBAAyB,QAAzB,C;QAYQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA1  
QyC,WA0QrC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA1QhB,OA4QO,W;O;KAXRX,C;8FAeA,yB;MAAA,0  
D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBA  
Ac,mBAA0B,QAA1B,C;QA0QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA3Q0C,WA2QtC,CAAY,OAA  
Z,CAAJ,EAA0B,OAA1B,C;;QA3QhB,OA6QO,W;O;KAZRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M  
A6QA,oC;MAAA,gC;MA7QA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc  
,mBAAuB,QAAvB,C;QA2QL,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAAY,aA5QuC,WA4QnC,CAAY,oBAA  
Z,CAAJ,EAA0B,oBAA1B,C;;QA5QhB,OA8QO,W;O;KA1RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M  
AAA,yD;QAUI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA  
6QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA9QoC,WA8QhC,CAAY,OAAZ,CAAJ,EA9QiD,cA8QvB,





OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aA  
AI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAGB,SAAhB,  
gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sG  
AGX,iD;MAWoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,  
cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAA  
A,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB  
,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CA  
Ab,C;;MAEhB,OAAO,W;K;sGAGX,yB;MAAA,oC;MAAA,gC;MAAA,wD;QAWoB,Q;QAAhB,wBAAGB,SAAh  
B,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAY,aAAI,oBAAJ,EAAa,cAAc,oBAAd,CAAb,C;;QAEhB,OA  
AO,W;O;KAdX,C;IAiBA,8C;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAL,I  
AAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAA  
Y,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;Q  
ACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,  
SAAb,M;QACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QA  
Aa,WAAA,SAAb,M;QACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SA  
Ab,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,  
wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB  
,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,  
gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,WAAY,WAAL,iBAAJ,C;;MAEhB,O  
AAO,W;K;IAGX,8B;MAII,OAAO,wBAAa,eAAW,YAAY,gBAAZ,CAAX,CAAb,C;K;IAGX,gC;MAII,OAAO,0B  
AAa,eAAc,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,CAAf,CAAb  
,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAa,YAAY,gBAAZ,CAAb,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eA  
Ac,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,CAAf,CAAb,C;K;IA  
GX,gC;MAII,OAAO,0BAAa,eAAgB,YAAY,gBAAZ,CAAhB,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAiB,  
YAAY,gBAAZ,CAAjB,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAc,YAAiB,eAAL,gBAAK,EAAa,GAAb,C  
AAjB,CAAd,CAAb,C;K;IAGX,2B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aAC  
A,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,qBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;  
MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,C  
AAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBA  
AN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SA  
AK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;a  
ACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6  
B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL  
,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gB  
AAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,S  
AAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K  
;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,  
6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAA  
L,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,g  
BAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,sBAAK,CAAL,EAAP,C;UAAL,K;;UACa,uBAAL  
,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,kC;MAII,OAAO,iBA Ae,aAAL,SAAK,CAAf,C;K;IAGX,oC;MAKiB,Q;M  
ADb,WAAW,iBAAGB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAL,IA  
AJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,C;MACX,wBAAa,SAAb,gB;Q  
AAa,WAAA,SAAb,M;QAAMB,IAAK,WAAL,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,i  
BA Ae,gBAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAL,IAAJ,C;;MACxB,OA  
AO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAGB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SA  
Ab,M;QAAMB,IAAK,WAAL,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,  
C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMB,IAAK,WAAL,IAAJ,C;;MACxB,OAAO,I;K;IAGX,







;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UA AU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MA AA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UAC C,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;Q AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,E AAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,S AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA EhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB, cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;K ARX,C;0FAWA,yB;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAA gB,cAAhB,UAAgB,SAAhB,O;UACI,WAAW,UAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB, OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cA AA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KA ZX,C;oFAeA,yB;MAAA,wE;MAiOA,+D;MAjOA,yC;QASW,kBAAU,oB;QAIoD,Q;QAAhB,iD;UAAgB,cAAhB, e;UACI,UAIoID,WakOvC,CAAY,OAAZ,C;UMv5UP,U;UADP,YNy5Ue,WMz5UH,WNy5UwB,GMz5UxB,C;U ACL,IAAI,aAAJ,C;YACH,aNu5UuC,gB;YAA5B,WMt5UX,aNs5UgC,GMt5UhC,EAAS,MAAT,C;YACA,e;;YAE A,c;;UNm5UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApOT,OAsOO,W;O;KA/OX,C;sFAYA,yB;MAAA,wE;MAso A,+D;MATOA,yC;QASW,kBAAU,oB;QAsOD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAvOoD,WAuO1C,CAAY ,OAAZ,C;UMx6UP,U;UADP,YN06Ue,WM16UH,WN06UwB,GM16UxB,C;UACL,IAAI,aAAJ,C;YACH,aNw6U uC,gB;YAA5B,WMv6UX,aNu6UgC,GMv6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNo6UA,iB;UACA,IAAK, WAAI,OAAJ,C;;QAzOT,OA2OO,W;O;KApPX,C;sFAYA,yB;MAAA,wE;MA2OA,+D;MA3OA,yC;QASW,kBA AU,oB;QA2OD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA5OqD,WA4O3C,CAAY,OAAZ,C;UMz7UP,U;UADP, YN27Ue,WM37UH,WN27UwB,GM37UxB,C;UACL,IAAI,aAAJ,C;YACH,aNy7UuC,gB;YAA5B,WMx7UX,aNw 7UgC,GMx7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNq7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA9OT,OAgP O,W;O;KAzPX,C;sFAYA,yB;MAAA,wE;MAgPA,+D;MAhPA,yC;QASW,kBAAU,oB;QAqPD,Q;QAAhB,iD;UA AgB,cAAhB,e;UACI,UAjPmD,WaiPzC,CAAY,OAAZ,C;UM18UP,U;UADP,YN48Ue,WM58UH,WN48UwB,G M58UxB,C;UACL,IAAI,aAAJ,C;YACH,aN08UuC,gB;YAA5B,WMz8UX,aNy8UgC,GMz8UhC,EAAS,MAAT,C; YACA,e;;YAEA,c;;UNs8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAnPT,OAqPO,W;O;KA9PX,C;sFAYA,yB;MA AA,wE;MAqPA,+D;MArPA,yC;QASW,kBAAU,oB;QAqPD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAtPoD,WA sp1C,CAAY,OAAZ,C;UM39UP,U;UADP,YN69Ue,WM79UH,WN69UwB,GM79UxB,C;UACL,IAAI,aAAJ,C;Y ACH,aN29UuC,gB;YAA5B,WM19UX,aN09UgC,GM19UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNu9UA,iB;U ACA,IAAK,WAAI,OAAJ,C;;QAxPT,OA0PO,W;O;KAnQX,C;sFAYA,yB;MAAA,wE;MA0PA,+D;MA1PA,yC;Q ASW,kBAAU,oB;QA0PD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA3PqD,WA2P3C,CAAY,OAAZ,C;UM5+UP, U;UADP,YN8+Ue,WM9+UH,WN8+UwB,GM9+UxB,C;UACL,IAAI,aAAJ,C;YACH,aN4+UuC,gB;YAA5B,WM 3+UX,aN2+UgC,GM3+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNw+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;Q A7PT,OA+PO,W;O;KAxQX,C;sFAYA,yB;MAAA,wE;MA+PA,+D;MA/PA,yC;QASW,kBAAU,oB;QA+PD,Q;Q AAhB,iD;UAAgB,cAAhB,e;UACI,UAhQsD,WAgQ5C,CAAY,OAAZ,C;UM7/UP,U;UADP,YN+/Ue,WM//UH,W N+/UwB,GM//UxB,C;UACL,IAAI,aAAJ,C;YACH,aN6/UuC,gB;YAA5B,WM5/UX,aN4/UgC,GM5/UhC,EAAS, MAAT,C;YACA,e;;YAEA,c;;UNy/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAIQT,OAoQO,W;O;KA7QX,C;sFAY A,yB;MAAA,wE;MAoQA,+D;MApQA,yC;QASW,kBAAU,oB;QAoQD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,U ArQuD,WAqQ7C,CAAY,OAAZ,C;UM9gVP,U;UADP,YNghVe,WMhhVH,WNghVwB,GMhhVxB,C;UACL,IAA I,aAAJ,C;YACH,aN8gVuC,gB;YAA5B,WM7gVX,aN6gVgC,GM7gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN 0gVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAvQT,OAYQO,W;O;KAIRX,C;sFAYA,yB;MAAA,wE;MAyQA,oC;M AAA,+D;MAAA,gC;MAzQA,yC;QASW,kBAAU,oB;QAYQD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,UA1QoD ,WA0Q1C,CAAY,oBAAZ,C;UM/hVP,U;UADP,YNiiVe,WMjiVH,WNiiVwB,GMjiVxB,C;UACL,IAAI,aAAJ,C; YACH,aN+hVuC,gB;YAA5B,WM9hVX,aN8hVgC,GM9hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2hVA,iB; UACA,IAAK,WAAI,oBAAJ,C;;QA5QT,OA8QO,W;O;KAvRX,C;sFAYA,yB;MAAA,wE;MA8QA,+D;MA9QA,y

D;QAUW,kBAAU,oB;QA8QD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA/QiD,WA+QvC,CAAY,OAAZ,C;UMjj  
VP,U;UADP,YNmjVe,WMnjVH,WNmjVwB,GMnjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNijVuC,gB;YAA5B,W  
MhjVX,aNgjVgC,GMhjVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6iVA,iB;UACA,IAAK,WAjRyD,cAiRrD,C  
AAe,OAAf,CAAJ,C;;QAJRT,OAmRO,W;O;KA7RX,C;uFAaA,yB;MAAA,wE;MAmRA,+D;MAmRA,yD;QAUW,  
kBAAU,oB;QAmRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UApRiD,WAoRvC,CAAY,OAAZ,C;UMnkVP,U;U  
ADP,YNqkVe,WMrkVH,WNqkVwB,GMrkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmkVuC,gB;YAA5B,WMIkVX  
,aNkkVgC,GMkVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+jVA,iB;UACA,IAAK,WAtRyD,cAsRrD,CAAe,O  
AAf,CAAJ,C;;QaIRT,OAwRO,W;O;KAlSX,C;uFAaA,yB;MAAA,wE;MAwRA,+D;MAxRA,yD;QAUW,kBAAU,  
oB;QAwRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAzRiD,WAYRvC,CAAY,OAAZ,C;UMriVP,U;UADP,YNul  
Ve,WMvIVH,WNulVwB,GMvIVxB,C;UACL,IAAI,aAAJ,C;YACH,aNqlVuC,gB;YAA5B,WMIpVX,aNolVgC,G  
MplVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNilVA,iB;UACA,IAAK,WA3RyD,cA2RrD,CAAe,OAAf,CAAJ,C  
;;QA3RT,OA6RO,W;O;KAvSX,C;uFAaA,yB;MAAA,wE;MA6RA,+D;MA7RA,yD;QAUW,kBAAU,oB;QA6RD,  
Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA9RiD,WA8RvC,CAAY,OAAZ,C;UMvmVP,U;UADP,YNymVe,WMz  
mVH,WNymVwB,GMzmVxB,C;UACL,IAAI,aAAJ,C;YACH,aNumVuC,gB;YAA5B,WMItmVX,aNsmVgC,GMt  
mVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmmVA,iB;UACA,IAAK,WAhSyD,cAgSrD,CAAe,OAAf,CAAJ,C;  
;QAhST,OAKSO,W;O;KA5SX,C;uFAaA,yB;MAAA,wE;MAkSA,+D;MAISA,yD;QAUW,kBAAU,oB;QAKSD,Q;  
QAAhB,iD;UAAgB,cAAhB,e;UACI,UAnSiD,WAmSvC,CAAY,OAAZ,C;UMznVP,U;UADP,YN2nVe,WM3nVH  
,WN2nVwB,GM3nVxB,C;UACL,IAAI,aAAJ,C;YACH,aNynVuC,gB;YAA5B,WMIxnVX,aNwnVgC,GMxnVhC,E  
AAS,MAAT,C;YACA,e;;YAEA,c;;UNqnVA,iB;UACA,IAAK,WArSyD,cAqSrD,CAAe,OAAf,CAAJ,C;;QArST,O  
AuSO,W;O;KAjTX,C;uFAaA,yB;MAAA,wE;MAuSA,+D;MAvSA,yD;QAUW,kBAAU,oB;QAUdS,Q;QAAhB,iD  
;UAAgB,cAAhB,e;UACI,UAxSiD,WAwSvC,CAAY,OAAZ,C;UM3oVP,U;UADP,YN6oVe,WM7oVH,WN6oVw  
B,GM7oVxB,C;UACL,IAAI,aAAJ,C;YACH,aN2oVuC,gB;YAA5B,WMIoVX,aNo0VgC,GM1oVhC,EAAS,MAA  
T,C;YACA,e;;YAEA,c;;UNuoVA,iB;UACA,IAAK,WA1SyD,cA0SrD,CAAe,OAAf,CAAJ,C;;QA1ST,OA4SO,W;  
O;KATX,C;uFAaA,yB;MAAA,wE;MA4SA,+D;MA5SA,yD;QAUW,kBAAU,oB;QA4SD,Q;QAAhB,iD;UAAgB,  
cAAhB,e;UACI,UA7SiD,WA6SvC,CAAY,OAAZ,C;UM7pVP,U;UADP,YN+pVe,WM/pVH,WN+pVwB,GM/pVx  
B,C;UACL,IAAI,aAAJ,C;YACH,aN6pVuC,gB;YAA5B,WMI5pVX,aN4pVgC,GM5pVhC,EAAS,MAAT,C;YACA,  
e;;YAEA,c;;UNypVA,iB;UACA,IAAK,WA/SyD,cA+SrD,CAAe,OAAf,CAAJ,C;;QA/ST,OAIto,W;O;KA3TX,C;  
uFAaA,yB;MAAA,wE;MAiTA,+D;MAjTA,yD;QAUW,kBAAU,oB;QAItd,Q;QAAhB,iD;UAAgB,cAAhB,e;UA  
CI,UAItd,WakTvC,CAAY,OAAZ,C;UM/qVP,U;UADP,YNirVe,WMjrVH,WNirVwB,GMjrVxB,C;UACL,IAAI  
,aAAJ,C;YACH,aN+qVuC,gB;YAA5B,WMI9qVX,aN8qVgC,GM9qVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN  
2qVA,iB;UACA,IAAK,WApTyD,cAoTrD,CAAe,OAAf,CAAJ,C;;QApTT,OAsTO,W;O;KAhUX,C;uFAaA,yB;M  
AAA,wE;MASTa,oC;MAAA,+D;MAAA,gC;MatTA,yD;QAUW,kBAAU,oB;QAsTD,Q;QAAhB,iD;UAAgB,cAA  
hB,oB;UACI,UAvTiD,WAutvC,CAAY,oBAAZ,C;UMjsVP,U;UADP,YNmsVe,WMnsVH,WNmsVwB,GMnsVx  
B,C;UACL,IAAI,aAAJ,C;YACH,aNisVuC,gB;YAA5B,WMIhsVX,aNgsVgC,GMhsVhC,EAAS,MAAT,C;YACA,e  
;;YAEA,c;;UN6rVA,iB;UACA,IAAK,WazTyD,cAyTrD,CAAe,oBAAf,CAAJ,C;;QAZTT,OA2TO,W;O;KArUX,C  
;wFAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,U  
AAU,YAAY,OAAZ,C;UMv5UP,U;UADP,YNy5Ue,WMz5UH,WNy5UwB,GMz5UxB,C;UACL,IAAI,aAAJ,C;Y  
ACH,aNu5UuC,gB;YAA5B,WMI5UX,aNs5UgC,GMt5UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNm5UA,iB;U  
ACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB  
,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UMx6UP,U;UADP,YN06Ue,W  
MI6UH,WN06UwB,GM16UxB,C;UACL,IAAI,aAAJ,C;YACH,aNw6UuC,gB;YAA5B,WMIv6UX,aNu6UgC,GMv  
6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNo6UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAd  
X,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UA  
CI,UAAU,YAAY,OAAZ,C;UMz7UP,U;UADP,YN27Ue,WM37UH,WN27UwB,GM37UxB,C;UACL,IAAI,aAAJ,  
C;YACH,aNy7UuC,gB;YAA5B,WMIx7UX,aNw7UgC,GMx7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNq7UA,i  
B;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QA  
AhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM18UP,U;UADP,YN48U  
e,WM58UH,WN48UwB,GM58UxB,C;UACL,IAAI,aAAJ,C;YACH,aN08UuC,gB;YAA5B,WMIz8UX,aNy8UgC,G

Mz8UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNs8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,YAAU,OAAZ,C;UM39UP,U;UADP,YN69Ue,WM79UH,WN69UwB,GM79UxB,C;UACL,IAAI,aAAJ,C;YACH,aN29UuC,gB;YAA5B,WM19UX,aN09UgC,GM19UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNu9UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,YAAU,OAAZ,C;UM5+UP,U;UADP,YN8+Ue,WM9+UH,WN8+UwB,GM9+UxB,C;UACL,IAAI,aAAJ,C;YACH,aN4+UuC,gB;YAA5B,WM3+UX,aN2+UgC,GM3+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNw+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,YAAU,OAAZ,C;UM7/UP,U;UADP,YN+/Ue,WM//UH,WN+/UwB,GM//UxB,C;UACL,IAAI,aAAJ,C;YACH,aN6/UuC,gB;YAA5B,WM5/UX,aN4/UgC,GM5/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNy/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,YAAU,OAAZ,C;UM9gVP,U;UADP,YNghVe,WMhhVH,WNghVwB,GMhhVxB,C;UACL,IAAI,aAAJ,C;YACH,aN8gVuC,gB;YAA5B,WM7gVX,aN6gVgC,GM7gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN0gVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,UAAU,YAAU,oBAAZ,C;UM/hVP,U;UADP,YNiiVe,WMjivH,WNiiVwB,GMjivxB,C;UACL,IAAI,aAAJ,C;YACH,aN+hVuC,gB;YAA5B,WM9hVX,aN8hVgC,GM9hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2hVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UMjjVP,U;UADP,YNmjVe,WMnjVH,WNmjVwB,GMnjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNijVuC,gB;YAA5B,WMhjVX,aNgjVgC,GMhjVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6iVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UMnkVP,U;UADP,YNqkVe,WMrkVH,WNqkVwB,GMrkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmkVuC,gB;YAA5B,WMlkVX,aNkkVgC,GMlkVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+jVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UMrlVP,U;UADP,YNulVe,WMvlVH,WNulVwB,GMvlVxB,C;UACL,IAAI,aAAJ,C;YACH,aNqlVuC,gB;YAA5B,WMplVX,aNolVgC,GMplVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNilVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UMvmVP,U;UADP,YNymVe,WMzmVH,WNymVwB,GMzmVxB,C;UACL,IAAI,aAAJ,C;YACH,aNumVuC,gB;YAA5B,WMtmVX,aNsmVgC,GMtmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmmVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UMznVP,U;UADP,YN2nVe,WM3nVH,WN2nVwB,GM3nVxB,C;UACL,IAAI,aAAJ,C;YACH,aNynVuC,gB;YAA5B,WMxnVX,aNwnVgC,GMxnVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNqnVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UM3oVP,U;UADP,YN6oVe,WM7oVH,WN6oVwB,GM7oVxB,C;UACL,IAAI,aAAJ,C;YACH,aN2oVuC,gB;YAA5B,WM1oVX,aN0oVgC,GM1oVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNuoVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UM7pVP,U;UADP,YN+pVe,WM/pVH,WN+pVwB,GM/pVxB,C;UACL,IAAI,aAAJ,C;YACH,aN6pVuC,gB;YAA5B,WM5pVX,aN4pVgC,GM5pVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNypVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAU,OAAZ,C;UM/qVP,U;UADP,YNirVe,WMjrVH,WNirVwB,GMjrVxB,C;UACL,IAAI,aAAJ,C;YACH,aN+qVuC,gB;YAA5B,WM9qVX,aN8qVgC,GM9qVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2qVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,oC;

MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;  
UACI,UAAU,YAAY,oBAAZ,C;UMjsVP,U;UADP,YNmsVe,WMnsVH,WNmsVwB,GMnsVxB,C;UACL,IAAI,a  
AAJ,C;YACH,aNisVuC,gB;YAA5B,WMhsVX,aNgsVgC,GMhsVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6rV  
A,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,kC;MAAA,4  
C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,Y;QAA6C,OAAgB,qBAAhB,oBAAGB,C;O;MATrE,iDA  
UQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;4EAca,yB;MAAA,gE;MAAA  
,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CA  
AU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa  
,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVh  
B,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAA  
b,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8  
EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,  
WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MA  
AA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,  
CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,e  
AAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QA  
hVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;Q  
AAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,  
C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UA  
CI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;M  
AAiVA,oC;MAAA,gC;MAjVA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,0B;UACI  
,WAAy,WAhViB,SAGVb,CAAU,iBAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;0FAUA,yB;MAAA,gE;M  
AAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAgHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,  
WAjHwB,SAiHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAJHhB,OAkHO,W;O;KAZHX,C;4  
FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAmHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,  
WAAb,e;UACI,WAAy,WApHwB,SAoHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QApHhB,O  
AqHO,W;O;KA5HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAsHP,gB;QADb,Y  
AAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WAvHwB,SAuHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IA  
AnB,CAAJ,C;;QAvHhB,OAwhO,W;O;KA/HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBA  
Ab,C;QAYHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WA1HwB,SA0HpB,EAAU,cAAV,E  
AAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA1HhB,OA2HO,W;O;KAIIX,C;4FAUA,yB;MAAA,gE;MAAA,uC;Q  
AOW,kBAaA,eAAa,gBAAb,C;QA4HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WA7HwB,  
SA6HpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA7HhB,OA8HO,W;O;KArIX,C;2FAUA,yB;  
MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QA+HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;U  
ACI,WAAy,WAhIwB,SAGIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAhIhB,OaiIO,W;O;K  
AxIX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAKIP,gB;QADb,YAAY,C;QACZ,i  
D;UAAa,WAAb,e;UACI,WAAy,WAnIwB,SAmIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA  
nIhB,OAoIO,W;O;KA3IX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAqIP,gB;QAD  
b,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WAtIwB,SAsIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IA  
AnB,CAAJ,C;;QAtIhB,OAuIO,W;O;KA9IX,C;4FAUA,yB;MAAA,gE;MAAuIA,oC;MAAA,gC;MAvIA,uC;QAOW,  
kBAaA,eAAa,gBAAb,C;QAwIP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UACI,WAAy,WAZIwB,SAyIp  
B,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAZIhB,OA0IO,W;O;KAjJX,C;wGAUA,yB;MAAA  
,+D;MAAA,uC;QAOW,kBAAoB,gB;QAKIEd,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAXkEmC,U;UAA  
A,cAVQ,SAUR,EAwkET,cAxkES,EAwkET,sBAxkES,WAwkEA,IAxkEA,W;YAA6C,6B;;QAVhF,OAwo,W;O;  
KAIBX,C;4GAUA,yB;MAAA,oD;QA+kEiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAXkEmC,U;UAAA  
,yBAwkET,cAxkES,EAwkET,sBAxkES,WAwkEA,IAxkEA,W;YAA6C,6B;;QACHf,OAAO,W;O;KARX,C;8FA  
WA,6C;MAQiB,UACiB,M;MAF9B,YAAY,C;MACZ,wBAaA,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAy,  
WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;gGAGX,6C;MAQiB,UAC







AO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q  
AAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,  
O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;M  
ACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,S  
AAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADh  
B,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,O  
AAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QA  
AGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUo  
B,Q;MADhB,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV  
,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAGB,SAAh  
B,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,  
yB;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADhB,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cA  
AhB,UAAgB,SAAhB,O;UAAsB,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;;QACpC,OAAO,W;O;KAXX,C;4FAc  
A,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,S  
AAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAGC,OAAhC,C;;MACpC,OAAO,W  
;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,  
cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAGC,OAAhC,C;;MACpC,  
OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAGB,SAAhB,g  
B;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAGC,OAAhC,C;  
;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAGB,  
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAGC,O  
AAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,w  
BAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,E  
AAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;  
MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,  
WAAAnB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB  
AAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,  
WAAmB,WAAAnB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;  
MACZ,kBAAkB,O;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAA  
U,sBAAV,WAAmB,WAAAnB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAA  
A,gD;QAYoB,UAA8B,M;QAF9C,YAAY,C;QACZ,kBAAkB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,U  
AAgB,SAAhB,O;UAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAGC,oBAAhC,C;;QACpC,  
OAAO,W;O;KAbX,C;wFagBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;Q  
ACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WAAxB,C;;QAE  
IB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;  
QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WAAxB,C;;Q  
AEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,  
O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WAAxB,C;;  
QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBA  
AAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WAA  
xB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kB  
AAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WAA  
xB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,k  
BAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,WA  
AxB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,  
kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAawB,W



AAxB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAA  
Y,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,sBAAl,YAAJ,EAAI,oBAAJ,QAA  
V,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAdX,C;sGaiBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,wB;QAC  
Z,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,  
WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,wB;QA  
CZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6  
B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,wB;Q  
ACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA  
6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,wB;Q  
ACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EA  
A6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,w  
B;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,  
EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAy,  
wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAj  
B,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YA  
AY,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,C  
AAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,  
YAAy,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,  
CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,oC;MA  
AA,gD;QAUI,YAAy,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAi  
B,sBAAl,KAAJ,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;oFAmBA,6B;MAIoB,Q;M  
AAhB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAA  
hB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,  
wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAA  
gB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAAgB,  
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAAgB,SA  
AhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,yB;MAAA,oC;MAAA,gC;MAAA,oC;QAI  
oB,Q;QAAhB,wBAAgB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UAsB,OAAO,oBAAP,C;;O;KAJ1B,C;  
kGAOA,6B;MAOiB,UAAa,M;MAD1B,YAAy,C;MACZ,wBAaA,SAAb,gB;QAaA,WAAa,SAAb,M;QAaMB,QA  
AO,cAAP,EAAO,sBAAP,WAGB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAy,C;MACZ,wBA  
Aa,SAAb,gB;QAaA,WAAa,SAAb,M;QAaMB,QAAO,cAAP,EAAO,sBAAP,WAGB,IAAhB,C;;K;oGAGvB,6B;  
MAOiB,UAAa,M;MAD1B,YAAy,C;MACZ,wBAaA,SAAb,gB;QAaA,WAAa,SAAb,M;QAaMB,QAAO,cAAP,E  
AAO,sBAAP,WAGB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAy,C;MACZ,wBAaA,SAAb,gB;  
QAaA,WAAa,SAAb,M;QAaMB,QAAO,cAAP,EAAO,sBAAP,WAGB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAA  
a,M;MAD1B,YAAy,C;MACZ,wBAaA,SAAb,gB;QAaA,WAAa,SAAb,M;QAaMB,QAAO,cAAP,EAAO,sBAAP,  
WAGB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAy,C;MACZ,wBAaA,SAAb,gB;QAaA,WAA  
A,SAAb,M;QAaMB,QAAO,cAAP,EAAO,sBAAP,WAGB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B  
,YAAy,C;MACZ,wBAaA,SAAb,gB;QAaA,WAAa,SAAb,M;QAaMB,QAAO,cAAP,EAAO,sBAAP,WAGB,IA  
AhB,C;;K;oGAGvB,yB;MAAA,oC;MAAA,gC;MAAA,oC;QA  
OiB,UAAa,M;QAD1B,YAAy,C;QACZ,wBAaA,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UAAmB,QAAO,cAAP,  
EAAO,sBAAP,WAGB,iBAAhB,C;;O;KAPvB,C;IAUA,wB;MAaiB,Q;MAFb,IApsLO,qBAAQ,CAosLf,C;QAae,  
MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;  
QACR,MmB/1aG,MAAO,KnB+1aE,GmB/1aF,EnB+1aO,CmB/1aP,C;;MnBi2ad,OAAO,G;K;IAGX,0B;MAaiB,Q;  
MAFb,IAxtLO,qBAAQ,CAwtLf,C;QAae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,  
CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB93aG,MAAO,KnB83aE,GmB93aF,EnB83aO,CmB93aP,C;;

MnBg4ad,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA1uLO,qBAAQ,CA0uLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApvLO,qBAAQ,CAovLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9vLO,qBAAQ,CA8vLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAxwLO,qBAAQ,CAwwLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAIxLO,qBAAQ,CAkxLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9xLO,qBAAQ,CA8xLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB5+aG,MAAO,KnB4+aE,GmB5+aF,EnB4+aO,CmB5+aP,C;;MnB8+ad,OAAO,G;K;IAGX,0B;MAAiB,Q;MAFb,IA1yLO,qBAAQ,CA0yLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAh4LO,qBAAQ,CAG4Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAN5LO,qBAAQ,CAM5Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAI6LO,qBAAQ,CAs6Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAz7LO,qBAAQ,CAY7Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA58LO,qBAAQ,CA48Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAI/LO,qBAAQ,CAk/Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAArgMO,qBAAQ,CAqgMf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,

cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,oC;MAAA,sC;QAWI,IAxhMO,qBAAQ,CAwhMf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;4FA2BA,yB;MAAA,8D;MAAA,sC;QAOI,IA/mMO,qBAAQ,CA+mMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9nMO,qBAAQ,CA8nMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA7oMO,qBAAQ,CA6oMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA5pMO,qBAAQ,CA4pMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA3qMO,qBAAQ,CA2qMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA1rMO,qBAAQ,CA0rMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAxtMO,qBAAQ,CAwtMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAvuMO,qBAAQ,CAvuMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sE;MAAA,8D;MmBt9bA,iB;MnBs9bA,sC;QAEiB,Q;QAFb,IAp0MO,qBAAQ,CAo0Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACf,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/9bG,MAAO,KnB+9bO,QmB/9bP,EnB+9biB,CmB/9bjB,C;;QnBi+bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB5+bA,iB;MnB4+bA,sC;QAEiB,Q;QAFb,IA11MO,qBAAQ,CAk1Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACf,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr/bG,MAAO,KnBq/bO,QmBr/bP,EnBq/biB,CmBr/bjB,C;;QnBu/b

d,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBlgcA,iB;MnBkgcA,sC;QAeiB,Q;QAFb,IAh2M  
O,qBAAQ,CAg2Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C  
AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3gcG,MAAO,KnB2gcO,QmB3gcP,EnB2gciB,  
CmB3gcjB,C;;QnB6gcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBxhcA,iB;MnBwhcA,sC;  
QAeiB,Q;QAFb,IA92MO,qBAAQ,CA82Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBjicG,MAAO,KnBiicO  
,QmBjicP,EnBiiciB,CmBjicjB,C;;QnBmicd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB9icA,  
iB;MnB8icA,sC;QAeiB,Q;QAFb,IA53MO,qBAAQ,CA43Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C  
AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvjcG  
,MAAO,KnBujcO,QmBvjcP,EnBujciB,CmBvjcjB,C;;QnByjcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MA  
AA,8D;MmBpkcA,iB;MnBokcA,sC;QAeiB,Q;QAFb,IA14MO,qBAAQ,CA04Mf,C;UAAe,MAAM,6B;QACrB,eA  
Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,  
C;UACR,WmB7kcG,MAAO,KnB6kcO,QmB7kcP,EnB6kciB,CmB7kcjB,C;;QnB+kcd,OAAO,Q;O;KAnBX,C;kFA  
sBA,yB;MAAA,sE;MAAA,8D;MmB1lcA,iB;MnB0lcA,sC;QAeiB,Q;QAFb,IAx5MO,qBAAQ,CAw5Mf,C;UAAe,  
MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS  
,UAAK,CAAL,CAAT,C;UACR,WmBnmcG,MAAO,KnBmmcO,QmBnmcP,EnBmmciB,CmBnmcjB,C;;QnBqmc  
,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBhncA,iB;MnBgncA,sC;QAeiB,Q;QAFb,IAt6MO  
,qBAAQ,CAs6Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CA  
AV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBzncG,MAAO,KnBzncO,QmBzncP,EnBznciB,C  
mBzncjB,C;;QnB2ncd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBtocA,iB;MnBs  
ocA,sC;QAeiB,Q;QAFb,IAp7MO,qBAAQ,CAo7Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EA  
AT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB/ocG,MAAO  
,KnB+ocO,QmB/ocP,EnB+ociB,CmB/ocjB,C;;QnBipcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8  
D;MmBvqcA,iB;MnBuqcA,sC;QAeiB,Q;QAFb,IA1gNO,qBAAQ,CA0gNf,C;UAAe,MAAM,6B;QACrB,eAAe,SA  
AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA  
CR,WmBhrcG,MAAO,KnBgrcO,QmBhrcP,EnBgrciB,CmBhrcjB,C;;QnBkrcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB  
;MAAA,sE;MAAA,8D;MmB7rcA,iB;MnB6rcA,sC;QAeiB,Q;QAFb,IAxhNO,qBAAQ,CAwhNf,C;UAAe,MAAM,  
6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,  
CAAL,CAAT,C;UACR,WmBtscG,MAAO,KnBsscO,QmBtscP,EnBssciB,CmBtscjB,C;;QnBwscd,OAAO,Q;O;KA  
nBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBntcA,iB;MnBmtcA,sC;QAeiB,Q;QAFb,IAtiNO,qBAAQ,CAsiNf,  
C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA  
AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5tcG,MAAO,KnB4tcO,QmB5tcP,EnB4tciB,CmB5tcjB,C;;QnB8tc  
d,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBzucA,iB;MnByucA,sC;QAeiB,Q;QAFb,IApjN  
O,qBAAQ,CAojNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C  
AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlvcG,MAAO,KnBkvcO,QmBlvcP,EnBkvcjB,  
CmBlvcjB,C;;QnBovcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB/vcA,iB;MnB+vcA,sC;Q  
AeiB,Q;QAFb,IAlkNO,qBAAQ,CAkkNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC  
F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxwcG,MAAO,KnBwwc  
O,QmBxwcP,EnBwwciB,CmBxwcjB,C;;QnB0wcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;M  
mBrxcA,iB;MnBqxcA,sC;QAeiB,Q;QAFb,IAhlNO,qBAAQ,CAglNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,U  
AAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,W  
mB9xcG,MAAO,KnB8xcO,QmB9xcP,EnB8xciB,CmB9xcjB,C;;QnBgycd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;M  
AAA,sE;MAAA,8D;MmB3ycA,iB;MnB2ycA,sC;QAeiB,Q;QAFb,IA9INO,qBAAQ,CA8INf,C;UAAe,MAAM,6B;  
QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA  
AL,CAAT,C;UACR,WmBpzcG,MAAO,KnBozcO,QmBpzcP,EnBozciB,CmBpzcjB,C;;QnBszcd,OAAO,Q;O;KAn  
BX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBj0cA,iB;MnBi0cA,sC;QAeiB,Q;QAFb,IA5mNO,qBAAQ,CA4mN  
f,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA  
AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB10cG,MAAO,KnB00cO,QmB10cP,EnB00ciB,CmB10cjB,C;;QnB

40cd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBv1cA,iB;MnBu1cA,sC;QAeiB,Q;QAFb,IA1nNO,qBAAQ,CA0nNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBh2cG,MAAO,KnBg2cO,QmBh2cP,EnBg2ciB,CmBh2cjB,C;;QnBk2cd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9sNO,qBAAQ,CA8sNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5tNO,qBAAQ,CA4tNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA1uNO,qBAAQ,CA0uNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxvNO,qBAAQ,CAwvNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAtpNO,qBAAQ,CAswNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApxNO,qBAAQ,CAoxNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAlyNO,qBAAQ,CAkyNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAhzNO,qBAAQ,CAgzNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9zNO,qBAAQ,CA8zNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MmBxidA,iB;MnBwidA,sC;QAaiB,Q;QAFb,IAp5NO,qBAAQ,CAo5Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/idG,MAAO,KnB+idO,QmB/idP,EnB+idiB,CmB/idjB,C;;QnBijdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB5jdA,iB;MnB4jdA,sC;QAaiB,Q;QAFb,IAh6NO,qBAAQ,CAg6Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnkdg,MAAO,KnBmkdO,QmBnkdp,EnBmkdiB,CmBnkdiB,C;;QnBqkdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhldA,iB;MnBgldA,sC;QAaiB,Q;QAFb,IA56NO,qBAAQ,CA46Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvldG,MAAO,KnBuldO,QmBvldP,EnBuldiB,CmBvldjB,C;;QnByldd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBpmdA,iB;MnBomdA,sC;QAaiB,Q;QAFb,IAx7NO,qBAAQ,CAw7Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3mdG,MAAO,KnB2mdO,QmB3mdp,EnB2mdiB,CmB3mdjB,C;;QnB6mdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBxndA,iB;MnBwndA,sC;QAaiB,Q;QAFb,IAp8NO,qBAAQ,CAo8Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/ndG,MAAO,KnB+ndO,QmB/ndP,EnB+ndiB,CmB/ndjB,C;;QnBiodd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB5odA,iB;MnB4odA,sC;QAaiB,Q;QAFb,IAh9NO,qBAAQ,CAg9Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnpdG,MAAO,KnBmpdO,QmBnpdp,EnBmpdiB,CmBnpdjB,C;

;QnBqpdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhqda,iB;MnBgqda,sC;QAaiB,Q;QAFb,IA59NO,qBAAQ,CA49Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvqdG,MAAO,KnBuqdO,QmBvqdp,EnBuqdiB,CmbvqdiB,C;;QnByqdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBprda,iB;MnBorda,sC;QAaiB,Q;QAFb,IAx+NO,qBAAQ,CAw+Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3rdG,MAAO,KnB2rdO,QmB3rdP,EnB2rdiB,Cmb3rdjB,C;;QnB6rdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MmBxsdA,iB;MnBwsdA,sC;QAaiB,Q;QAFb,IAp/NO,qBAAQ,CAo/Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB/sdG,MAAO,KnB+sdO,QmB/sdP,EnB+sdiB,Cmb/sdjB,C;;QnBitdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBvudA,iB;MnBuudA,sC;QAaiB,Q;QAFb,IAxkOO,qBAAQ,CAwkOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9udG,MAAO,KnB8udO,QmB9udP,EnB8udiB,Cmb9udjB,C;;QnBgvdD,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB3vdA,iB;MnB2vdA,sC;QAaiB,Q;QAFb,IAplOO,qBAAQ,CAo/Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlwdG,MAAO,KnBkwdO,QmBlwdP,EnBkwdiB,CmbLwdjB,C;;QnBowdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB/wdA,iB;MnB+wdA,sC;QAaiB,Q;QAFb,IAhmOO,qBAAQ,CAGmOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBtxdG,MAAO,KnBsxdO,QmBtxdP,EnBsxdIb,CmbTxdjB,C;;QnBwxdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBnydA,iB;MnBmydA,sC;QAaiB,Q;QAFb,IA5mOO,qBAAQ,CA4mOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB1ydG,MAAO,KnB0ydO,QmB1ydP,EnB0ydiB,Cmb1yjdjB,C;;QnB4ydd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBvzdA,iB;MnBuzdA,sC;QAaiB,Q;QAFb,IAxnOO,qBAAQ,CAwnOf,C;UAAe,OA AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9zdG,MAAO,KnB8zdO,QmB9zdP,EnB8zdiB,Cmb9zdiB,C;;QnBg0dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB30dA,iB;MnB20dA,sC;QAaiB,Q;QAFb,IApoOO,qBAAQ,CAooOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBl1dG,MAAO,KnBk1dO,QmBl1dP,EnBk1diB,CmbL1djb,C;;QnBo1dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB/1dA,iB;MnB+1dA,sC;QAaiB,Q;QAFb,IAhpOO,qBAAQ,C AgpOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBt2dG,MAAO,KnBs2dO,QmBt2dP,EnBs2diB,CmbT2djB,C;;Q nBw2dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBn3dA,iB;MnBm3dA,sC;QAaiB,Q;QAFb,IA5pOO,q BAAQ,CA4pOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB13dG,MAAO,KnB03dO,QmB13dP,EnB03diB,Cmb1 3djB,C;;QnB43dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MmBv4dA,iB;MnBu4dA,sC;QAaiB ,Q;QAFb,IAxqOO,qBAAQ,CAwqOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B; QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB94dG,MAAO,KnB84dO,QmB 94dP,EnB84diB,Cmb94djB,C;;QnBg5dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;Q AFb,IA1vOO,qBAAQ,CA0vOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb, aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA AW,C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtwOO,qBAAQ,C AswOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI, QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA AW,C;;QAGnB,OAAO, Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAIxOO,qBAAQ,CAkxOf,C;UAAe,OAAO,I ;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA AL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA AW,C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA, yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9xOO,qBAAQ,CA8xOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,U AAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IA

AI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,s  
C;QAWiB,Q;QAFb,IA1yOO,qBAAQ,CA0yOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA  
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1  
zOO,qBAAQ,CAszOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C  
AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;  
QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1000,qBAAQ,CAk0Of,C  
;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S  
AAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAj  
BX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9000,qBAAQ,CA80Of,C;UAAe,OAAO,I;QACtB,  
eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAA  
T,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAA  
A,oC;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1100,qBAAQ,CA01Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,  
sBAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAK,CAAL,EAAT,C;UACR,  
IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAA  
A,8D;MAAA,kD;QAaiB,Q;QAFb,IAh700,qBAAQ,Cag7Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C  
AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA  
AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;  
0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA9700,qBAAQ,CA87Of,C;UAAe,MAAM,6B;Q  
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA  
L,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QA  
GnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA5800,qBAAQ,C  
A48Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA  
CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,C  
AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAai  
B,Q;QAFb,IA1900,qBAAQ,CA09Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+  
B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EA  
akB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;  
MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAx+OO,qBAAQ,CAw+Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA  
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA  
I,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAn  
BX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAt/OO,qBAAQ,CAs/Of,C;UAAe,MAAM,  
6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,  
CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;  
;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IApgPO,qBAA  
Q,CAogPf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;  
UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC  
,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QA  
aiB,Q;QAFb,IAlhPO,qBAAQ,CAkhPf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,  
+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EA  
AkB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,s  
E;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAhiPO,qBAAQ,CAGiPf,C;UAAe,MAAM,6B;QACrB,eAAe  
,SAAS,sBAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAK,CAAL,EAAT,C  
;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAA  
O,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IApnPO,qBAAQ,CAonPf,C;UAAe,OAA  
O,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,  
CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;  
;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAhoPO,qBAAQ,CAGoPf,

C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5oPO,qBAAQ,CA4oPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAxpPO,qBAAQ,CAwpPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IApQPO,qBAAQ,CAoqPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAhrPO,qBAAQ,CAgrPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5rPO,qBAAQ,CA4rPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAxsPO,qBAAQ,CAwsPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5tPO,qBAAQ,CAotPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IAtyPO,qBAAQ,CAsyPf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBj8eG,MAAO,KnBi8eE,GmBj8eF,EnBi8eO,CmBj8eP,C;;MnBm8ed,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAtzPO,qBAAQ,CAszPf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB59eG,MAAO,KnB49eE,GmB59eF,EnB49eO,CmB59eP,C;;MnB89ed,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAp0PO,qBAAQ,CAo0Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KA AJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA10PO,qBAAQ,CA00Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAh1PO,qBAAQ,CAg1Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAAt1PO,qBAAQ,CAAt1Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAp2PO,qBAAQ,CAo2Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBljFg,MAAO,KnBkjfE,GmBljff,EnBkjfO,CmBljFP,C;;MnBojfd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA52PO,qBAAQ,CA42Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBvjfG,MAAO,KnBujfE,GmBvjfF,EnBujfO,CmBvjfP,C;;MnByjfd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA12PO,qBAAQ,CA02Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,wC;MAWiB,Q;MAFb,IA57PO,qBAAQ,CA47Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MA



Ab,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAt8PO,qBAAQ,CAs8Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAh9PO,qBAAQ,CAg9Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA19PO,qBAAQ,CA09Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAp+PO,qBAAQ,CAo+Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA9+PO,qBAAQ,CA8+Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAx/PO,qBAAQ,CAw/Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAIlgQO,qBAAQ,CAkgQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAI1QO,qBAAQ,CA01Qf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhmQO,qBAAQ,CAgmQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA5gQO,qBAAQ,CA4gQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IA11QO,qBAAQ,CA01Qf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA11QO,qBAAQ,CA01Qf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA5mQO,qBAAQ,CA4mQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAInQO,qBAAQ,CAknQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAxnQO,qBAAQ,CAwnQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA9nQO,qBAAQ,CA8nQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IApoQO,qBAAQ,CAooQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA1oQO,qBAAQ,CA0oQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,wB;MAaiB,Q;MAFb,IA9tQO,qBAAQ,CA8tQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBrqfG,MAAO,KnBqqfE,GmBrqfF,EnBqqfO,CmBrqfP,C;;MnBuqfd,OAAO,G;K;I

AGX,0B;MAaiB,Q;MAFb,IAIvQO,qBAAQ,CAkvQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBpsfG,MAAO,KnBosfE,GmBpsfF,EnBosfO,CmBpsfP,C;;MnBssfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApwQO,qBAAQ,CAowQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9wQO,qBAAQ,CA8wQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAxxQO,qBAAQ,CAwxQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAlyQO,qBAAQ,CAkyQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA5yQO,qBAAQ,CA4yQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IAxzQO,qBAAQ,CAwzQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBlzfG,MAAO,KnBkzfE,GmBlzfF,EnBkzfO,CmBlzfP,C;;MnBozfd,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IAp0QO,qBAAQ,CAo0Qf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB3zfG,MAAO,KnB2zfE,GmB3zfF,EnB2zfO,CmB3zfP,C;;MnB6zfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAt0QO,qBAAQ,CAs0Qf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA15QO,qBAAQ,CA05Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA76QO,qBAAQ,CA66Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAh8QO,qBAAQ,CAg8Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAI5QO,qBAAQ,CA55Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAI9QO,qBAAQ,CA99Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAz/QO,qBAAQ,Cay/Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA5gRO,qBAAQ,CA4gRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;

YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,s  
C;QAWI,IA/hRO,qBAAQ,CA+hRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,S  
AAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OA  
Aa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,  
C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAA  
A,oC;MAAA,sC;QAWI,IAIjRO,qBAAQ,CAkjRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBA  
AqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aA  
AU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,  
CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;4FA2BA,yB;MAAA,8D;M  
AAA,sC;QAOI,IAzoRO,qBAAQ,CAyoRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAA  
L,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,  
OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA  
AJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QA  
OI,IAxpRO,qBAAQ,CAwpRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;  
QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SA  
Ab,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,  
UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAvqRO,  
qBAAQ,CAuqRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI  
,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,Q  
AAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YA  
CV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAtrRO,qBAAQ,CA  
srRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB  
,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,  
CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,  
C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IArsRO,qBAAQ,CAqsRf,C;UAAe,  
OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,  
OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UA  
CR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,O  
AAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAptRO,qBAAQ,CAotRf,C;UAAe,OAAO,I;QAC  
tB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC  
3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SA  
AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA  
pBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IANuRO,qBAAQ,CAmuRf,C;UAAe,OAAO,I;QACtB,cAAc,UA  
AK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,S  
AAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,  
C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8F  
AuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAlvRO,qBAAQ,CAkvRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,  
C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,  
C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IA  
AI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;M  
AAA,8D;MAAA,oC;MAAA,sC;QAOI,IAjwRO,qBAAQ,CAiwRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,  
C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAA  
T,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IA  
AI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;M  
AAA,sE;MAAA,8D;MmB5xgBA,iB;MnB4xgBA,sC;QAEiB,Q;QAFb,IA91RO,qBAAQ,CA81Rf,C;UAAe,MAA  
M,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACf,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA  
K,CAAL,CAAT,C;UACR,WmBrygBG,MAAO,KnBqygBO,QmBrygBP,EnBqygBiB,CmBrygBjB,C;;QnBuygBd,O  
AAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBlzgBA,iB;MnBkzgBA,sC;QAEiB,Q;QAFb,IA52RO

,qBAAQ,CA42Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3zgBG,MAAO,KnB2zgBO,QmB3zgBP,EnB2zgBiB,CmB3zgBjB,C;;QnB6zgBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBx0gBA,iB;MnBw0gBA,sC;QAeiB,Q;QAFb,IA13RO,qBAAQ,CA03Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj1gBG,MAAO,KnBi1gBO,QmBj1gBP,EnBi1gBiB,CmBj1gBjB,C;;QnBm1gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB91gBA,iB;MnB81gBA,sC;QAeiB,Q;QAFb,IAx4RO,qBAAQ,CAw4Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBv2gBG,MAAO,KnBu2gBO,QmBv2gBP,EnBu2gBiB,CmBv2gBjB,C;;QnBy2gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBp3gBA,iB;MnBo3gBA,sC;QAeiB,Q;QAFb,IA5RO,qBAAQ,CA5Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB73gBG,MAAO,KnB63gBO,QmB73gBP,EnB63gBiB,CmB73gBjB,C;;QnB+3gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB14gBA,iB;MnB04gBA,sC;QAeiB,Q;QAFb,IAp6RO,qBAAQ,CAo6Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBn5gBG,MAAO,KnBm5gBO,QmBn5gBP,EnBm5gBiB,CmBn5gBjB,C;;QnBq5gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBh6gBA,iB;MnBg6gBA,sC;QAeiB,Q;QAFb,IAI7RO,qBAAQ,CAk7Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz6gBG,MAAO,KnBy6gBO,QmBz6gBP,EnBy6gBiB,CmBz6gBjB,C;;QnB26gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBt7gBA,iB;MnBs7gBA,sC;QAeiB,Q;QAFb,IAh8RO,qBAAQ,CAg8Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/7gBG,MAAO,KnB+7gBO,QmB/7gBP,EnB+7gBiB,CmB/7gBjB,C;;QnBi8gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmB58gBA,iB;MnB48gBA,sC;QAeiB,Q;QAFb,IA98RO,qBAAQ,CA88Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBr9gBG,MAAO,KnBq9gBO,QmBr9gBP,EnBq9gBiB,CmBr9gBjB,C;;QnBu9gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB7+gBA,iB;MnB6+gBA,sC;QAeiB,Q;QAFb,IApiSO,qBAAQ,CAoiSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBt/gBG,MAAO,KnBs/gBO,QmBt/gBP,EnBs/gBiB,CmBt/gBjB,C;;QnBw/gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBnghBA,iB;MnBmghBA,sC;QAeiB,Q;QAFb,IALjSO,qBAAQ,CAkjSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5ghBG,MAAO,KnB4ghBO,QmB5ghBP,EnB4ghBiB,CmB5ghBjB,C;;QnB8ghBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBzhhBA,iB;MnByhhBA,sC;QAeiB,Q;QAFb,IAhkSO,qBAAQ,CAgkSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlihBG,MAAO,KnBkihBO,QmBlihBP,EnBkihBiB,CmBlihBjB,C;;QnBoihBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB/ihBA,iB;MnB+ihBA,sC;QAeiB,Q;QAFb,IA9kSO,qBAAQ,CA8kSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxjhBG,MAAO,KnBwjhBO,QmBxjhBP,EnBwjhBiB,CmBxjhBjB,C;;QnB0jhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBrkhBA,iB;MnBqkhBA,sC;QAeiB,Q;QAFb,IA5ISO,qBAAQ,CA4ISf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9khBG,MAAO,KnB8khBO,QmB9khBP,EnB8khBiB,CmB9khBjB,C;;QnBglhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB3lhBA,iB;MnB2lhBA,sC;QAeiB,Q;QAFb,IA1mSO,qBAAQ,CA0mSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpmhBG,MAAO,KnBomhBO,QmBpmhBP,EnBomhBiB,CmBpmhBjB,C;;QnBsmhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBjnhBA,iB;MnBinhBA,sC;QAeiB,Q;QAFb,IAxnSO,qBAAQ,CAwnSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U

ACR,WmB1nhBG,MAAO,KnB0nhBO,QmB1nhBP,EnB0nhBiB,CmB1nhBjB,C;;QnB4nhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBvohBA,iB;MnBvohBA,sC;QAeiB,Q;QAFb,IAtoSO,qBAAQ,CAsoSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhphBG,MAAO,KnBgphBO,QmBhphBP,EnBgphBiB,CmBhphBjB,C;;QnBkphBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmB7phBA,iB;MnB6phBA,sC;QAeiB,Q;QAFb,IApSO,qBAAQ,CAopSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBtqhBG,MAAO,KnBsqhBO,QmBtqhBP,EnBsqhBiB,CmBtqhBjB,C;;QnBwqhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxuSO,qBAAQ,CAwuSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAtvSO,qBAAQ,CAsvSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApwSO,qBAAQ,CAowSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAIxSO,qBAAQ,CAkxSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9ySO,qBAAQ,CA8ySf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5zSO,qBAAQ,CA4zSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA10SO,qBAAQ,CA00Sf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAx1SO,qBAAQ,CAw1Sf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MmB92hBA,iB;MnB82hBA,sC;QAaiB,Q;QAFb,IA96SO,qBAAQ,CA86Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr3hBG,MAAO,KnBq3hBO,QmBr3hBP,EnBq3hBiB,CmBr3hBjB,C;;QnBu3hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB14hBA,iB;MnBk4hBA,sC;QAaiB,Q;QAFb,IA17SO,qBAAQ,CA07Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz4hBG,MAAO,KnBy4hBO,QmBz4hBP,EnBy4hBiB,CmBz4hBjB,C;;QnB24hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBt5hBA,iB;MnBs5hBA,sC;QAaiB,Q;QAFb,IAt8SO,qBAAQ,CAs8Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB75hBG,MAAO,KnB65hBO,QmB75hBP,EnB65hBiB,CmB75hBjB,C;;QnB+5hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB16hBA,iB;MnB06hBA,sC;QAaiB,Q;QAFb,IAI9SO,qBAAQ,CAk9Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj7hBG,MAAO,KnBi7hBO,QmBj7hBP,EnBi7hBiB,CmBj7hBjB,C;;QnBm7hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB97hBA,iB;MnB87hBA,sC;QAaiB,Q;QAFb,IA99SO,qBAAQ,CA89Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,U

AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr8hBG,MAAO,KnBq8hBO,QmBr8hBP,EnBq8hBiB,CmBr8hBjB,C;;QnBu8hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB19hBA,iB;MnBk9hBA,sC;QAaiB,Q;QAFb,IA1+SO,qBAAQ,CA0+Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz9hBG,MAAO,KnBy9hBO,QmBz9hBP,EnBy9hBiB,CmBz9hBjB,C;;QnB29hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBt+hBA,iB;MnBs+hBA,sC;QAaiB,Q;QAFb,IAt/SO,qBAAQ,CAs/Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB7+hBG,MAAO,KnB6+hBO,QmB7+hBP,EnB6+hBiB,CmB7+hBjB,C;;QnB++hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB1/hBA,iB;MnB0/hBA,sC;QAaiB,Q;QAFb,IAIlgTO,qBAAQ,CAkgTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBjgiBG,MAAO,KnBigiBO,QmBjgiBP,EnBigiBiB,CmBjgiBjB,C;;QnBmgiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MmB9giBA,iB;MnB8giBA,sC;QAaiB,Q;QAFb,IA9gTO,qBAAQ,CA8gTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBrhiBG,MAAO,KnBqhiBO,QmBrhiBP,EnBqhiBiB,CmBrhiBjB,C;;QnBuhiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB7iiBA,iB;MnB6iiBA,sC;QAaiB,Q;QAFb,IAImTO,qBAAQ,CAkmTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpjiBG,MAAO,KnBojiBO,QmBpjiBP,EnBojiBiB,CmBpjiBjB,C;;QnBsjiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBjkiBA,iB;MnBikiBA,sC;QAaiB,Q;QAFb,IA9mTO,qBAAQ,CA8mTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxkiBG,MAAO,KnBwkiBO,QmBxkiBP,EnBwkiBiB,CmBxkiBjB,C;;QnB0kiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBrliBA,iB;MnBqliBA,sC;QAaiB,Q;QAFb,IA1nTO,qBAAQ,CA0nTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5liBG,MAAO,KnB4liBO,QmB5liBP,EnB4liBiB,CmB5liBjB,C;;QnB8liBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBzmiBA,iB;MnBymiBA,sC;QAaiB,Q;QAFb,IAtoTO,qBAAQ,CAsTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhniBG,MAAO,KnBgniBO,QmBhniBP,EnBgniBiB,CmBhniBjB,C;;QnBkniBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB7niBA,iB;MnB6niBA,sC;QAaiB,Q;QAFb,IAIpTO,qBAAQ,CAkpTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpoiBG,MAAO,KnBooiBO,QmBpoiBP,EnBooiBiB,CmBpoiBjB,C;;QnBsoiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBjpiBA,iB;MnBipiBA,sC;QAaiB,Q;QAFb,IA9pTO,qBAAQ,CA8pTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxpiBG,MAAO,KnBwpiBO,QmBxpiBP,EnBwpiBiB,CmBxpiBjB,C;;QnB0piBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBrqiBA,iB;MnBqqiBA,sC;QAaiB,Q;QAFb,IA1qTO,qBAAQ,CA0qTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5qiBG,MAAO,KnB4qiBO,QmB5qiBP,EnB4qiBiB,CmB5qiBjB,C;;QnB8qiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBzriBA,iB;MnByriBA,sC;QAaiB,Q;QAFb,IAtrTO,qBAAQ,CAsrTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhsiBG,MAAO,KnBgsiBO,QmBhsiBP,EnBgsiBiB,CmBhsiBjB,C;;QnBksiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MmB7siBA,iB;MnB6siBA,sC;QAaiB,Q;QAFb,IAIsTO,qBAAQ,CAksTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBptiBG,MAAO,KnBotiBO,QmBptiBP,EnBotiBiB,CmBptiBjB,C;;QnBstiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IApxTO,qBAAQ,CAoxTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAhyTO,qBAAQ,CAgyTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAA



AU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA9oUO,qBAAQ,CA8oUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1pUO,qBAAQ,CA0pUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1rUO,qBAAQ,CAkrUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1sUO,qBAAQ,CA0sUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1tUO,qBAAQ,CAsqUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1uUO,qBAAQ,CAkuUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1vUO,qBAAQ,CAkvUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1wUO,qBAAQ,CAkwUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1xUO,qBAAQ,CAkxUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1yUO,qBAAQ,CAkyUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1zUO,qBAAQ,CAkzUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA2oUO,qBAAQ,CAo0Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBvwjBG,MAAO,KnBuwjBE,GmBvwjBF,EnBuwjBO,CmBvwjBP,C;;MnBywjBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAh0UO,qBAAQ,CAG0Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBlyjBG,MAAO,KnBkyjBE,GmBlyjBF,EnBkyjBO,CmBlyjBP,C;;MnBoyjBd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA91UO,qBAAQ,CA81Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAp2UO,qBAAQ,CAo2Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA12UO,qBAAQ,CA02Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA13UO,qBAAQ,CA3Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA93UO,qBAAQ,CA83Uf,C;QAAe,OAAO,I;MAcTb,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBx3jBG,MAAO,KnBw3jBE,GmBx3jBF,EnBw3jB



O,CmBx3jBP,C;;MnB03jBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAt4UO,qBAAQ,CAs4Uf,C;QA Ae,OAAO,I; MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB 73jBG,MAAO,KnB63jBE,GmB73jBF,EnB63jBO,CmB73jBP,C;;MnB+3jBd,OAAO,G;K;IAGX,gC;MAOiB,Q;M A Fb,IAp4UO,qBAAQ,CAo4Uf,C;QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CA A V,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IA GX,wC;MAWiB,Q;MAFb,IAt9UO,qBAAQ,CAs9Uf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MAC G,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,C AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAh+UO,qBAAQ, CAg+Uf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ, UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;; MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA1+UO,qBAAQ,CA0+Uf,C;QA Ae,MAAM,6B;MACrB,UAA U,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAA Q,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q; MA Fb,IAp/UO,qBAAQ,CAo/Uf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU, CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA9/UO,qBAAQ,CA8/Uf,C;QA Ae,MA AM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QAC R,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;I AGX,0C;MAWiB,Q;MAFb,IAxgVO,qBAAQ,CAwgVf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;M ACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAA b,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAlhVO,qBAA Q,CAkhVf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAA Q,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM, C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA5hVO,qBAAQ,CA4hVf,C;QA Ae,MAAM,6B;MACrB,UA AU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SA AQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB, Q;MAFb,IAtiVO,qBAAQ,CAsiVf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU ,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CA AjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IApnVO,qBAAQ,CAonVf,C;QA Ae, OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QA CR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K ;IAGX,gD;MAOiB,Q;MAFb,IA1nVO,qBAAQ,CA0nVf,C;QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MAC G,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,C AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhoVO,qBAAQ, CAgoVf,C;QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,U AAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;; MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAtoVO,qBAAQ,CAsoVf,C;QA Ae,OAAO,I;MACtB,UAAU,UA AK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GA AR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb ,IA5oVO,qBAAQ,CA4oVf,C;QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,i B;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UA AoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAlpVO,qBAAQ,CAkpVf,C;QA Ae,OAAO,I;M A CtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,U AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD; MAOiB,Q;MAFb,IAxpVO,qBAAQ,CAwpVf,C;QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAA b,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA 6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA9pVO,qBAAQ,CA8pVf,C; QA Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL

,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA  
O,G;K;IAGX,gD;MAOiB,Q;MAFb,IApqVO,qBAAQ,CAoqVf,C;QAAe,OAAO,I;MACTb,UAAU,UAAK,CAAL,C  
;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,c  
AAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,yB;MAMI,OAnvVO,qBAAQ,C;  
K;IASvVnB,2B;MAMI,OAqvVO,qBAAQ,C;K;IAuvVnB,2B;MAMI,OArvVO,qBAAQ,C;K;IAwvVnB,2B;MAMI,  
OAtvVO,qBAAQ,C;K;IAyvVnB,2B;MAMI,OAyvVO,qBAAQ,C;K;IA0vVnB,2B;MAMI,OAxvVO,qBAAQ,C;K;I  
A2vVnB,2B;MAMI,OAzvVO,qBAAQ,C;K;IA4vVnB,2B;MAMI,OA1vVO,qBAAQ,C;K;IA6vVnB,2B;MAMI,OA  
3vVO,qBAAQ,C;K;gFA8vVnB,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB  
,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;gFAGX,gC;MAMoB,Q;MAAhB,wBAAGB  
,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO  
,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAA  
V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAG  
B,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;M  
AMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAw  
B,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,  
M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB  
,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MAC  
rD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,U  
AAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QA  
MoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UAsB,IAAI,UAAU,oBAAV,CAAJ,  
C;YAAwB,OAAO,K;;QACrD,OAAO,I;O;KAPX,C;kFAUA,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAs  
B,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,  
C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;  
oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAM  
mC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD  
;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,  
e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,  
OAAP,C;;MAArC,gB;K;oFAGJ,yB;MAAA,oC;MAAA,gC;MAAA,oC;QAMmC,Q;QAAhB,iD;UAGB,cAAhB,0  
B;UAsB,OAAO,oBAAP,C;;QAArC,gB;O;KANJ,C;gGASA,6B;MAn4KiB,gB;MADb,YAAY,C;MACZ,iD;QAA  
a,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA44KnB,gB;K;kGAGJ,6B;MAr4KiB,gB;  
MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA84Kn  
B,gB;K;kGAGJ,6B;MAv4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sB  
AAP,WAAgB,IAAhB,C;;MAg5KnB,gB;K;kGAGJ,6B;MAz4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;  
QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAk5KnB,gB;K;kGAGJ,6B;MA34KiB,gB;MADb,YA  
AY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAo5KnB,gB;K;kG  
AGJ,6B;MA74KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAg  
B,IAAhB,C;;MA55KnB,gB;K;kGAGJ,6B;MA/4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QA  
AO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAw5KnB,gB;K;kGAGJ,6B;MAj5KiB,gB;MADb,YAAY,C;MAC  
Z,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA05KnB,gB;K;kGAGJ,yB;MA  
AA,6B;MAAA,sC;MA15KA,oC;MAAA,gC;MA05KA,2BASiB,yB;QAn6KjB,oC;QAAA,gC;eAm6KiB,0B;UAAA  
,4B;YAAE,aAAe,c;YA55KjB,gB;YADb,YAAY,C;YACZ,iD;cAAa,WAAb,0B;cAAmB,QAAO,cAAP,EAAO,sBA  
AP,WAAgB,iBAAhB,C;;YA45KmB,W;W;S;OAAzB,C;MATjB,oC;QAn5KiB,gB;QADb,YAAY,C;QACZ,iD;UA  
Aa,WAAb,0B;UAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA45KnB,gB;O;KATJ,C;kFAYA,yB;  
MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAImWO,qBAAQ,CAkmWf,C;UACI,MAAM,mCAA8B,+  
BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EA  
AuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;Q  
AgBqB,Q;QAHjB,IAhnWO,qBAAQ,CAgnWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,C  
AAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAuB,UAAK,KAAL,CAAvB,C;;QA

EIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9nWO,qBAAQ,CA8nWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5oWO,qBAAQ,CA4oWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1pWO,qBAAQ,CA0pWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxqWO,qBAAQ,CAwqWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtrWO,qBAAQ,CAsrWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IApsWO,qBAAQ,CAosWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAItWO,qBAAQ,CAktWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KAnBX,C;gGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxyWO,qBAAQ,CAwyWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtzWO,qBAAQ,CAszWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp0WO,qBAAQ,CAo0Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAI1WO,qBAAQ,CAk1Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAh2WO,qBAAQ,CAg2Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA92WO,qBAAQ,CA82Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA53WO,qBAAQ,CA43Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA14WO,qBAAQ,CA04Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAx5WO,qBAAQ,CAw5Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;4GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9+WO,qBAAQ,CA8+Wf,C;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5/WO,

qBAAQ,CA4/Wf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1gXO,qBAAQ,CA0gXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxhXO,qBAAQ,CAwhXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtiXO,qBAAQ,CAsiXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAiXO,qBAAQ,CAkXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAikXO,qBAAQ,CAkkXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhlXO,qBAAQ,CAglXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IA9lXO,qBAAQ,CA8lXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IArrXO,qBAAQ,CAqrXf,C;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IApsXO,qBAAQ,CAosXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAntXO,qBAAQ,CAmtXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAluXO,qBAAQ,CAkuXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAjvXO,qBAAQ,CAivXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAhwXO,qBAAQ,CAgwXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA9wXO,qBAAQ,CA9wXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA8xXO,qBAAQ,CA8xXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAIbqB,Q;QAHjB,IA7yXO,qBAAQ,CA6yXf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KApBX,C;4FAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE6B,UAE0,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EA AwB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,U

AEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB, UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAL,sBAAJ,S AAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B ,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAA kB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAL,sBAAJ ,SAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE 0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kB AAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAL,sB AAJ,SAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC; QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB ,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAL,sBAAJ,SAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,u C;QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QAC rB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EA AL,sBAAJ,SAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAA A,uC;QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;Q ACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ, EAAL,sBAAJ,SAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;M AAA,oC;MAAA,gC;MAAA,uC;QAE0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAA M,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI, cAAc,oBAAU,sBAAI,cAAJ,EAAL,sBAAJ,UAAV,EAawB,wBAAxB,E;;QAEIB,OAAO,W;O;KAnBX,C;0GAsBA ,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE6B,Q;QAFzB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAA M,mCAA8B,+BAA9B,C;QACrB,kBAAqB,UAAI,YAAJ,EAAL,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI, cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C; 4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAA e,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C ;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAp BX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C ;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CA AhB,C;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W; O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OA AO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;Q AEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QA CIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd ,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB; QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QA CIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd ,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAE0B,Q;QAFtB,YA AY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAL ,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,iB,UAAI,KAAJ,CAAjB,EAA6B,W AA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,oC;MAAA,gC;M AAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACr B,kBAAkB,UAAI,YAAJ,EAAL,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAAV,EAAL,iB,s



pBX,C;0GAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAA Y,wB;QACZ ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CA AhB,C;UACI,cAAc,oBAAU,sBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O;KAp BX,C;4FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAgBoB,Q;QAHhB,IAtRZO,qBAAQ,CAsrZf,C;UAAe,OA AO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB10oBO,W;QIB20o BP,kBAaKB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;8FAuBA,yB;MAAA,gD;MAAA,gE;MAA A,gD;QAIBoB,Q;QAHhB,IAtsZO,qBAAQ,CAszZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBA AO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB12oBO,W;QIBm2oBP,kBAaKB,O;QACIB,wBAAGB,SAAhB,gB;UA AgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,O AAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAtZO,qBAAQ,CAsTz f,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB13oB O,W;QIB23oBP,kBAaKB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAA V,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;M AAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAtuZO,qBAAQ,CAsuZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBA AvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB15oBO,W;QIBm5oBP,kBAaKB,O;QACIB,wBAAGB ,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WA AJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAtvZO ,qBAAQ,CAsvZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;Q AA5C,akB16oBO,W;QIB26oBP,kBAaKB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAA c,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB ;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAtwZO,qBAAQ,CAswZf,C;UAAe,OAAO,OAAO,OAAP ,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB18oBO,W;QIBm8oBP,kBAaKB,O;QA CIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MA AO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q; QAHhB,IAtxZO,qBAAQ,CAsxZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb, C;QAA+B,8B;QAA5C,akB19oBO,W;QIB29oBP,kBAaKB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAh B,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX ,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAtyZO,qBAAQ,CAsyZf,C;UAAe,OAAO ,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB1/oBO,W;QIBm/oBP,k BAaKB,O;QACIB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB ,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,o C;MAAA,gC;MAAA,gD;QAIBoB,Q;QAHhB,IAtzZO,qBAAQ,CAszZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,k BAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB1gpBO,W;QIB2gpBP,kBAaKB,O;QACIB,wBA AgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MA AO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0GAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IA94Z O,qBAAQ,CA84Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B ;QAA5C,akB1ipBO,W;QIBmipBP,kBAaKB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,U AAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,gD; MAAA,gE;MAAA,gD;QAEI,IA/5ZO,qBAAQ,CA+5Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa, mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB3jpBO,W;QIB4jpBP,kBAaKB,O;QACIB,wD;UACI,cAAc,UAA U,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O; KAtBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IAh7ZO,qBAAQ,CAg7Zf,C;UAAe,OAAO,OAA O,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBp1pBO,W;QIBq1pBP,kBAaK B,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,W AAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IAj8ZO,qBA AQ,CAi8Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5 C,akB7mpBO,W;QIB8mpBP,kBAaKB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAA





6pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,0BAAV,EAAuB,sBAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;8GAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QACI,I A9yaO,qBAAQ,CA8yaf,C;UAAe,OAAO,W;QACtB,sBAAqB,UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,g BAAb,C;QAA2B,sBAAI,aAAJ,C;QAAXC,akBn8pBO,W;QIBo8pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gB AAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OA AO,M;O;KArBX,C;gHAWBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA5zaO,qBAAQ,CA4zaf,C;UAAe,OA AO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C; QAA3C,akBz9pBO,W;QIB09pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB ,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;M AAA,qD;MAAA,gE;MAAA,uC;QAYI,IA10aO,qBAAQ,CA00af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CA AL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akB/+pBO,W;QIBg/pBP,iB AAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;U ACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QA YI,IAx1aO,qBAAQ,CAw1af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACkC,kBAArB,eA Ae,gBAAf,C;QAA6B,sBAAI,aAAJ,C;QAA1C,akBrgqBO,W;QIBsgqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI, gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,O AAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAt2aO,qBAAQ,CAs2af,C;UAAe,O AAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ, C;QAA3C,akB3hqBO,W;QIB4hqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAA jB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB; MAAA,qD;MAAA,gE;MAAA,uC;QAYI,Iap3aO,qBAAQ,CAo3af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,C AAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akBjjqBO,W;QIBkjqBP,iB AAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;U ACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QA YI,IAI4aO,qBAAQ,CAk4af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAA kB,gBAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,akBvkqBO,W;QIBwkqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UA CI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAE X,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAh5aO,qBAAQ,CAG5af,C;UA Ae,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACsC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,a AAJ,C;QAA9C,akB7lqBO,W;QIB8lqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB, aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA ,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAYI,IA95aO,qBAAQ,CA85af,C;UAAe,OAAO,W ;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA 3C,akBnnqBO,W;QIBonqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,KAAV,EAAiB,0BAAjB,E AA8B,sBAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;8EAsBA,yB;M A/zBA,gD;MAAA,gE;MA+zBA,gD;QAcW,sB;;UA7zBS,Q;UAHhB,IAtZ0,qBAAQ,CAsrZf,C;YAAe,qBAAO,O Ag0BH,OA0h0BG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+zBzB,OA/zByB, C;UAA5C,akB10oBO,W;UIB20oBP,kBA8zBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cA4zBwB,SA5zBV,C AAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAyzBP,yB;O;KADJ,C;gF AiBA,yB;MAzzBA,gD;MAAA,gE;MAyzBA,gD;QAEW,sB;;UAvzBS,Q;UAHhB,IAtsZ0,qBAAQ,CAssZf,C;YAA e,qBAAO,OA0zBH,OA1zBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB, OAzzByB,C;UAA5C,akB12oBO,W;UIBm2oBP,kBAwzBmB,O;UAvzBnB,iD;YAAgB,cAAhB,e;YACI,cAszBwB, SAtzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAmzBP,yB;O; KAFJ,C;gFAkBA,yB;MANzBA,gD;MAAA,gE;MANzBA,gD;QAEW,sB;;UAjzBS,Q;UAHhB,IAttZ0,qBAAQ,CAs tZf,C;YAAe,qBAAO,OAozBH,OApzBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B, sBAmzBzB,OAnzByB,C;UAA5C,akB13oBO,W;UIB23oBP,kBAkzBmB,O;UAjzBnB,iD;YAAgB,cAAhB,e;YACI, cAgzBwB,SAhzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA6 yBP,yB;O;KAFJ,C;gFAkBA,yB;MA7yBA,gD;MAAA,gE;MA6yBA,gD;QAEW,sB;;UA3yBS,Q;UAHhB,IAtuZ0,q

BAAQ,CAsuZf,C;YAAe,qBAAO,OA8yBH,OA9yBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6yBzB,OA7yByB,C;UAA5C,akB15oBO,W;UIBm5oBP,kBA4yBmB,O;UA3yBnB,iD;YAAgB,cAAhB,e;YACI,cA0yBwB,SA1yBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAuyBP,yB;O;KAFJ,C;gFAkBA,yB;MAvyBA,gD;MAAA,gE;MAuyBA,gD;QAeW,sB;;UARYBS,Q;UAHhB,IAtwZO,qBAAQ,CAsvZf,C;YAAe,qBAAO,OAwyBH,OAxyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuyBzB,OAvyByB,C;UAA5C,akB16oBO,W;UIB26oBP,kBA5yBmB,O;UARYBnB,iD;YAAgB,cAAhB,e;YACI,cAoyBwB,SApyBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAiyBP,yB;O;KAFJ,C;gFAkBA,yB;MAjyBA,gD;MAAA,gE;MAiyBA,gD;QAeW,sB;;UA/xBS,Q;UAHhB,IAtwZO,qBAAQ,CAswZf,C;YAAe,qBAAO,OAkyBH,OAlyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAiyBzB,OAjyByB,C;UAA5C,akB18oBO,W;UIBm8oBP,kBAgyBmB,O;UA/xBnB,iD;YAAgB,cAAhB,e;YACI,cA8xBwB,SA9xBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA2xBP,yB;O;KAFJ,C;gFAkBA,yB;MA3xBA,gD;MAAA,gE;MA2xBA,gD;QAeW,sB;;UAzxBS,Q;UAHhB,IAtxZO,qBAAQ,CAsxZf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2xBzB,OA3xByB,C;UAA5C,akB19oBO,W;UIB29oBP,kBA0xBmB,O;UAzxBnB,iD;YAAgB,cAAhB,e;YACI,cAwxBwB,SAxxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAqxBP,yB;O;KAFJ,C;gFAkBA,yB;MARxBA,gD;MAAA,gE;MAqxBA,gD;QAeW,sB;;UANxBS,Q;UAHhB,IAtyZO,qBAAQ,CAsyZf,C;YAAe,qBAAO,OAxBH,OAxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAqxBzB,OArxByB,C;UAA5C,akB1/oBO,W;UIBm/oBP,kBAoxBmB,O;UANxBnB,iD;YAAgB,cAAhB,e;YACI,cAkxBwB,SAIxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA+wBP,yB;O;KAFJ,C;gFAkBA,yB;MA/wBA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MA+wBA,gD;QAeW,sB;;UA7wBS,Q;UAHhB,IAtzZO,qBAAQ,CAszZf,C;YAAe,qBAAO,OAxBH,OAxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+wBzB,OA/wByB,C;UAA5C,akB1gpBO,W;UIB2gpBP,kBA8wBmB,O;UA7wBnB,iD;YAAgB,cAAhB,OB;YACI,cA4wBwB,SA5wBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAyWP,yB;O;KAFJ,C;4FAkBA,yB;MAzwBA,gD;MAAA,gE;MAywBA,gD;QAeW,6B;;UA1wBP,IA94ZO,qBAAQ,CA84Zf,C;YAAe,4BAAO,OA0wBI,OA1wBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAywBIB,OAzwBkB,C;UAA5C,akB1ipBO,W;UIBmipBP,kBAwwB0B,O;UAvwB1B,wD;YACI,cA5wB+B,SAtwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAmwBP,gC;O;KAFJ,C;8FAkBA,yB;MANwBA,gD;MAAA,gE;MAMwBA,gD;QAgBW,6B;;UApwBP,IA5ZO,qBAAQ,CA+5Zf,C;YAAe,4BAAO,OAowBI,OApwBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAmwBIB,OAAnwBkB,C;UAA5C,akB3jpBO,W;UIB4jpBP,kBAkwB0B,O;UAjwB1B,wD;YACI,cAgwB+B,SAhwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA6vBP,gC;O;KAhBJ,C;8FAMBA,yB;MA7vBA,gD;MAAA,gE;MA6vBA,gD;QAgBW,6B;;UA9vBP,IAh7ZO,qBAAQ,CAg7Zf,C;YAAe,4BAAO,OA8vBI,OA9vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6vBIB,OA7vBkB,C;UAA5C,akB1plpBO,W;UIB1qlpBP,kBA4vB0B,O;UA3vB1B,wD;YACI,cA0vB+B,SA1vBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAuvBP,gC;O;KAhBJ,C;8FAMBA,yB;MAvvBA,gD;MAAA,gE;MAuvBA,gD;QAgBW,6B;;UAxvBP,IAj8ZO,qBAAQ,CAi8Zf,C;YAAe,4BAAO,OAwwBI,OAxxvBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuvBIB,OAvvBkB,C;UAA5C,akB7mpBO,W;UIB8mpBP,kBA5vB0B,O;UARvB1B,wD;YACI,cAovB+B,SApvBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAivBP,gC;O;KAhBJ,C;8FAMBA,yB;MAjvBA,gD;MAAA,gE;MAivBA,gD;QAgBW,6B;;UALvBP,IAI9ZO,qBAAQ,CAk9Zf,C;YAAe,4BAAO,OAkvBI,OA1vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAivBIB,OAjvBkB,C;UAA5C,akBtopBO,W;UIBuopBP,kBAgvB0B,O;UA/uB1B,wD;YACI,cA8uB+B,SA9uBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA2uBP,gC;O;KAhBJ,C;8FAMBA,yB;MA3uBA,gD;MAAA,gE;MA2uBA,gD;QAgBW,6B;;UA5uBP,IAN+ZO,qBAAQ,CAM+Zf,C;YAAe,4BAAO,OA4uBI,OA5uBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2uBIB,OA3uBkB,C;UAA5C,akB/ppBO,W;UIBgpBP,kBA0uB0B,O;UAzuB1B,wD;YACI,cAwuB+B,SAxuBjB,CAAU,KAAV,E

AAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAquBP,g  
C;O;KAhBJ,C;8FamBA,yB;MAruBA,gD;MAAA,gE;MAquBA,gD;QAgBW,6B;;UAtuBP,IAp/ZO,qBAAQ,CAo/  
Zf,C;YAAe,4BAAO,OAsuBI,OAtuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sB  
AquBlB,OArubkB,C;UAA5C,akBxrpBO,W;UlByrpBP,kBAOuB0B,O;UAnuB1B,wD;YACI,cAkuB+B,SaluBjB,C  
AAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,  
M;;;QA+tBP,gC;O;KAhBJ,C;8FamBA,yB;MA/tBA,gD;MAAA,gE;MA+tBA,gD;QAgBW,6B;;UAhuBP,IArGO,q  
BAAQ,CAqgaf,C;YAAe,4BAAO,OAgubI,OAhuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,  
C;UAA+B,sBA+tBIB,OA/tBkB,C;UAA5C,akBjtpBO,W;UlBktpBP,kBA8tB0B,O;UA7tB1B,wD;YACI,cA4tB+B,S  
A5tBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX  
,4BAAO,M;;;QAytBP,gC;O;KAhBJ,C;8FamBA,yB;MAztBA,gD;MAAA,gE;MAAA,oC;MAytBA,gD;QAgBW,6  
B;;UA1tBP,IAthaO,qBAAQ,CAshaf,C;YAAe,4BAAO,OA0tBI,OA1tBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mB  
AAO,CAAP,IAAb,C;UAA+B,sBAytBIB,OAztBkB,C;UAA5C,akB1upBO,W;UlB2upBP,kBAwtB0B,O;UAvtB1B,  
wD;YACI,cAstB+B,SAttBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,EAA9B,C;YACd,MAAO,W  
AAI,WAAJ,C;;UAEX,4BAAO,M;;;QamtBP,gC;O;KAhBJ,C;gFamBA,+B;MAOoB,Q;MADhB,UAAe,C;MACf,  
wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAG  
X,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,  
OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAG  
B,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UA  
Ae,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO  
,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YA  
AO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAh  
B,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;  
MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;  
MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAh  
B,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAO  
oB,Q;QADhB,UAAe,C;QACf,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,YAAO,SAAS,oB  
AAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;4FAaA,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB  
,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADh  
B,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OA  
AO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q  
ACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,  
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q  
;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MA  
EX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAh  
B,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wB  
AAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MA  
OoB,Q;MADhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,  
C;;MAEX,OAAO,G;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACIB,wBA  
AgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C;;QAEJ,OAAO,G;O;KAVX,  
C;gFAaA,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OA  
AO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,  
gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADh  
B,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,O  
AAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;  
QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAg  
B,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,  
Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;  
MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAGB,cAAA,S

AAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,yB; MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACpB,wBAAgB,SAAhB,gB;UAAgB,cAAhB, UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C;;QAEJ,OAAO,G;O;KAbX,C;kFAGBA,+B;MAUoB,Q;MADh B,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,O AAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QA CI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB, SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MA UoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CA AP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA, SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;M ACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;m FAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,S AAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB; QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,oC;MAA A,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACf,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;U ACI,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;Q AUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C ;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAg B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAG BA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh B,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB, sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CA AP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wB AAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;m FAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,S AAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MA TpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAA T,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QAC A,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAb X,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB, SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX, C;mFAGBA,yB;MGjyrBA,6B;MHyrBA,sC;QAWoB,Q;QADhB,UGjyrBmC,cHyrBnB,CGjyrBmB,C;QHkyrBnC, wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG/msBiD,cH+msBjD,GG/msB2D,KAAK,GH+msBzD,S AAS,OAAT,CG/msBoE,KAAX,IAAf,C;;QHinsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGlzrBA,6B;MHkzrBA,s C;QAWoB,Q;QADhB,UGlzrBmC,cHkzrBnB,CGlzrBmB,C;QHmzrBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SA AhB,M;UACI,MGhosBiD,cHgosBjD,GGhosB2D,KAAK,GHgosBzD,SAAS,OAAT,CGhosBoE,KAAX,IAAf,C;;Q HkosBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGn0rBA,6B;MHm0rBA,sC;QAWoB,Q;QADhB,UGn0rBmC,cHm0 rBnB,CGn0rBmB,C;QH0rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGjpsBiD,cHipsBjD,GG jpsB2D,KAAK,GHipsBzD,SAAS,OAAT,CGjpsBoE,KAAX,IAAf,C;;QHmpsBrD,OAAO,G;O;KAdX,C;mFAiBA, yB;MGp1rBA,6B;MH01rBA,sC;QAWoB,Q;QADhB,UGp1rBmC,cHo1rBnB,CGp1rBmB,C;QHq1rBnB,wBAAgB, SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGlqsBiD,cHkqsBjD,GGlqsB2D,KAAK,GHkqsBzD,SAAS,OAAT, CGlqsBoE,KAAX,IAAf,C;;QHoqsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGr2rBA,6B;MHq2rBA,sC;QAWoB, Q;QADhB,UGr2rBmC,cHq2rBnB,CGr2rBmB,C;QHs2rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UA CI,MGnrsBiD,cHmrsBjD,GGnrsB2D,KAAK,GHmrsBzD,SAAS,OAAT,CGnrsBoE,KAAX,IAAf,C;;QHqrsBrD,O AAO,G;O;KAdX,C;mFAiBA,yB;MGt3rBA,6B;MHs3rBA,sC;QAWoB,Q;QADhB,UGt3rBmC,cHs3rBnB,CGt3rB mB,C;QH3rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGpssBiD,cHossBjD,GGpssB2D,KAA K,GHossBzD,SAAS,OAAT,CGpssBoE,KAAX,IAAf,C;;QHsssBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGv4rBA,

6B;MHu4rBA,sC;QAWoB,Q;QADhB,UGv4rBmC,cHu4rBnB,CGv4rBmB,C;QHw4rBnB,wBAAgB,SAAhB,gB;U  
AAgB,cAAA,SAAhB,M;UACI,MGrtsBiD,cHqtsBjD,GGrtsB2D,KAAK,GHqtsBzD,SAAS,OAAT,CGrtsBoE,KAA  
X,IAAf,C;;QHutsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGx5rBA,6B;MHw5rBA,sC;QAWoB,Q;QADhB,UGx5  
rBmC,cHw5rBnB,CGx5rBmB,C;QH5rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGtusBiD,c  
HsusBjD,GGtusB2D,KAAK,GHsusBzD,SAAS,OAAT,CGtusBoE,KAAK,IAAf,C;;QHwusBrD,OAAO,G;O;KAdX  
,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MGz6rBA,6B;MHy6rBA,sC;QAWoB,Q;QADhB,UGz6rBmC,cHy6rBnB,C  
Gz6rBmB,C;QH06rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,MGvvsBiD,cHuvsBjD,  
GGvvsB2D,KAAK,GHuvsBzD,SAAS,oBAAT,CGvvsBoE,KAAK,IAAf,C;;QHvvsBrD,OAAO,G;O;KAdX,C;mFA  
iBA,yB;MoBv7rBA,+B;MpBu7rBA,sC;QAWoB,Q;QADhB,UoBt7rBqC,eAAW,oBpBs7rB/B,CoBt7rB+B,CAAX,  
C;QpBu7rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBrwsBmD,epBqwsBnD,GoBrwsB8D,K  
AAK,KpBqwsB5D,SAAS,OAAT,CoBrwsBuE,KAAK,CAAhB,C;;QpBuwsBvD,OAAO,G;O;KAdX,C;mFAiBA,y  
B;MoBx8rBA,+B;MpBw8rBA,sC;QAWoB,Q;QADhB,UoBv8rBqC,eAAW,oBpBu8rB/B,CoBv8rB+B,CAAX,C;Q  
pBw8rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBtxsBmD,epBsxsBnD,GoBtxsB8D,KAAK,  
KpBsxsB5D,SAAS,OAAT,CoBtxsBuE,KAAK,CAAhB,C;;QpBwxsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoBz  
9rBA,+B;MpBy9rBA,sC;QAWoB,Q;QADhB,UoBx9rBqC,eAAW,oBpBw9rB/B,CoBx9rB+B,CAAX,C;QpBy9rBr  
C,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBvysBmD,epBuysBnD,GoBvysB8D,KAAK,KpBuys  
B5D,SAAS,OAAT,CoBvysBuE,KAAK,CAAhB,C;;QpByysBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoB1+rBA,+  
B;MpB0+rBA,sC;QAWoB,Q;QADhB,UoBz+rBqC,eAAW,oBpBy+rB/B,CoBz+rB+B,CAAX,C;QpB0+rBrC,wBA  
AgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBxzsBmD,epBwzsBnD,GoBxzsB8D,KAAK,KpBwzsB5D,S  
AAS,OAAT,CoBxzsBuE,KAAK,CAAhB,C;;QpB0zsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB3/rBA,+B;MpB  
2/rBA,sC;QAWoB,Q;QADhB,UoB1/rBqC,eAAW,oBpB0/rB/B,CoB1/rB+B,CAAX,C;QpB2/rBrC,wBAAgB,SAAh  
B,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBz0sBmD,epBy0sBnD,GoBz0sB8D,KAAK,KpBy0sB5D,SAAS,OAAT,  
CoBz0sBuE,KAAK,CAAhB,C;;QpB20sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB5gsBA,+B;MpB4gsBA,sC;Q  
AWoB,Q;QADhB,UoB3gsBqC,eAAW,oBpB2gsB/B,CoB3gsB+B,CAAX,C;QpB4gsBrC,wBAAgB,SAAhB,gB;U  
AAgB,cAAA,SAAhB,M;UACI,MoB11sBmD,epB01sBnD,GoB11sB8D,KAAK,KpB01sB5D,SAAS,OAAT,CoB11  
sBuE,KAAK,CAAhB,C;;QpB41sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB7hsBA,+B;MpB6hsBA,sC;QAWoB  
,Q;QADhB,UoB5hsBqC,eAAW,oBpB4hsB/B,CoB5hsB+B,CAAX,C;QpB6hsBrC,wBAAgB,SAAhB,gB;UAAgB,c  
AAA,SAAhB,M;UACI,MoB32sBmD,epB22sBnD,GoB32sB8D,KAAK,KpB22sB5D,SAAS,OAAT,CoB32sBuE,K  
AAK,CAAhB,C;;QpB62sBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoB9isBA,+B;MpB8isBA,sC;QAWoB,Q;QADh  
B,UoB7isBqC,eAAW,oBpB6isB/B,CoB7isB+B,CAAX,C;QpB8isBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh  
B,M;UACI,MoB53sBmD,epB43sBnD,GoB53sB8D,KAAK,KpB43sB5D,SAAS,OAAT,CoB53sBuE,KAAK,CAAh  
B,C;;QpB83sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MoB/jsBA,+B;MpB+jsBA,sC;QAWo  
B,Q;QADhB,UoB9jsBqC,eAAW,oBpB8jsB/B,CoB9jsB+B,CAAX,C;QpB+jsBrC,wBAAgB,SAAhB,gB;UAAgB,c  
AAhB,UAAgB,SAAhB,O;UACI,MoB74sBmD,epB64sBnD,GoB74sB8D,KAAK,KpB64sB5D,SAAS,oBAAT,CoB  
74sBuE,KAAK,CAAhB,C;;QpB+4sBvD,OAAO,G;O;KAdX,C;IAiBA,mC;MAIoB,UAMT,M;MANP,wBAAgB,S  
AAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,eAAJ,C;UACI,MAAM,gCAAYB,2BAAwB,SAAXB,MAAZB,C;;  
;MAId,OAAO,0D;K;wFAGX,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,g  
B;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WA  
AI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoB  
A,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB  
,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAA  
O,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6  
B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh  
B,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,  
OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;Q  
AFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,O  
AAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EA  
AY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,a

AAa,gB;QACb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAA  
M,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0  
FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAA Y,gB;QACZ,aAAa,gB;QACb,wBAAGB,S  
AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,  
MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MA  
AA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAA Y,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAAgB,cAAA,S  
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;Q  
AGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB  
,Q;QAFhB,YAA Y,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAA  
U,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,E  
AAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;Q  
AFhB,YAA Y,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,  
UAAU,oBAAV,CAAJ,C;YACI,KAAM,WAAI,oBAAJ,C;;YAEN,MAAO,WAAI,oBAAJ,C;;;QAGf,OAAO,cAAK,  
KAAL,EAAY,MAAZ,C;O;KAjBX,C;IAoBA,+B;MakGI,WmBpgtBO,MAAO,KnBogtBG,gBmBpgtBH,EnBy6sB  
H,KA2FkB,OmBpgtBf,C;MnBqgtBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IA  
AK,WA9FqB,GA8FP,UAAK,CAAL,CA9FO,EAAnB,KA8FqB,CAAM,CAAN,CA9FF,CA8FrB,C;;MA9FT,OAGG  
O,I;K;IA7FX,iC;MAwGI,WmBphtBO,MAAO,KnBohtBG,gBmBphtBH,EnBm7sBH,KAiGkB,OmBphtBf,C;MnBq  
htBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApGqB,GAoGP,UAAK,C  
AAL,CAPGO,EAAnB,KAoGqB,CAAM,CAAN,CAPGF,CAoGrB,C;;MApGT,OAsGO,I;K;IANGX,iC;MA8GI,Wm  
BpitBO,MAAO,KnBoitBG,gBmBpitBH,EnB67sBH,KAuGkB,OmBpitBf,C;MnBqitBd,WAAW,iBAAa,IAAb,C;M  
ACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1GqB,GA0GP,UAAK,CAAL,CA1GO,EAAnB,KA0GqB,  
CAAM,CAAN,CA1GF,CA0GrB,C;;MA1GT,OA4GO,I;K;IAzGX,iC;MAoHI,WmBpjtBO,MAAO,KnBojtBG,gBm  
BpjtBH,EnBu8sBH,KA6GkB,OmBpjtBf,C;MnBqjtBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IA  
AIB,M;QACI,IAAK,WAhHqB,GAgHP,UAAK,CAAL,CAhHO,EAAnB,KAhHqB,CAAM,CAAN,CAhHF,CAGhr  
B,C;;MAhHT,OAKHO,I;K;IA/GX,iC;MA0HI,WmBpktBO,MAAO,KnBoktBG,gBmBpktBH,EnBi9sBH,KAmHkB,  
OmBpktBf,C;MnBqktBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtHq  
B,GAsHP,UAAK,CAAL,CAtHO,EAAnB,KAsHqB,CAAM,CAAN,CAtHF,CAsHrB,C;;MAiHT,OAWHO,I;K;IARh  
X,iC;MAGII,WmBpltBO,MAAO,KnBoltBG,gBmBpltBH,EnB29sBH,KAyHkB,OmBpltBf,C;MnBqltBd,WAAW,i  
BAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5HqB,GA4HP,UAAK,CAAL,CA5HO,  
EAAnB,KA4HqB,CAAM,CAAN,CA5HF,CA4HrB,C;;MA5HT,OA8HO,I;K;IA3HX,iC;MASII,WmBpmtBO,MAA  
O,KnBomtBG,gBmBpmtBH,EnBq+sBH,KA+HkB,OmBpmtBf,C;MnBqmtBd,WAAW,iBAAa,IAAb,C;MACX,aA  
AU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAlIqB,GAKIP,UAAK,CAAL,CAIIO,EAAnB,KAKIqB,CAAM,CAA  
N,CAIIF,CAKIrB,C;;MAIIT,OAoIO,I;K;IAjIX,iC;MA4II,WmBpntBO,MAAO,KnBontBG,gBmBpntBH,EnB++sB  
H,KAqIkB,OmBpntBf,C;MnBqntBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAA  
K,WAxIqB,GAWIP,UAAK,CAAL,CAXIO,EAAnB,KAwIqB,CAAM,CAAN,CAXIF,CAwIrB,C;;MAXIT,OA0IO,I;  
K;IAvIX,iC;MAKJI,WmBpotBO,MAAO,KnBootBG,gBmBpotBH,EnBy/sBH,KA2IkB,OmBpotBf,C;MnBqotBd,W  
AAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9IqB,GA8IP,sBAAK,CAAL,EA  
9IO,EAAnB,KA8IqB,CAAM,CAAN,CA9IF,CA8IrB,C;;MA9IT,OAGJO,I;K;8EA7IX,yB;MAAA,gE;MmBlgtBA,i  
B;MnBkgtBA,8C;QAQI,WmBpgtBO,MAAO,KnBogtBG,gBmBpgtBH,EnBogtBS,KAAM,OmBpgtBf,C;QnBqgtB  
d,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CA  
AV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MmBlhtBA,i  
B;MnBkhtBA,8C;QAQI,WmBphtBO,MAAO,KnBohtBG,gBmBphtBH,EnBohtBS,KAAM,OmBphtBf,C;QnBqhtB  
d,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CA  
AV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlitBA,i  
B;MnBkitBA,8C;QAQI,WmBpitBO,MAAO,KnBoitBG,gBmBpitBH,EnBoitBS,KAAM,OmBpitBf,C;QnBqitBd,W  
AAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,  
EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MmBljtBA,iB;M  
nBkjtBA,8C;QAQI,WmBpjtBO,MAAO,KnBojtBG,gBmBpjtBH,EnBojtBS,KAAM,OmBpjtBf,C;QnBqjtBd,WAA

W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA  
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlktBA,iB;MnB  
kktBA,8C;QAQI,WmBpktBO,MAAO,KnBoktBG,gBmBpktBH,EnBoktBS,KAAM,OmBpktBf,C;QnBqktBd,WAA  
W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA  
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlItBA,iB;MnBk  
ItBA,8C;QAQI,WmBpltBO,MAAO,KnBoltBG,gBmBpltBH,EnBoltBS,KAAM,OmBpltBf,C;QnBqltBd,WAAW,e  
AAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA  
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlmtBA,iB;MnBkmt  
BA,8C;QAQI,WmBpmtBO,MAAO,KnBomtBG,gBmBpmtBH,EnBomtBS,KAAM,OmBpmtBf,C;QnBqmtBd,WA  
AW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,E  
AAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlntBA,iB;Mn  
BkntBA,8C;QAQI,WmBpntBO,MAAO,KnBontBG,gBmBpntBH,EnBontBS,KAAM,OmBpntBf,C;QnBqntBd,WA  
AW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,E  
AAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MmBlo  
tBA,iB;MnBkotBA,8C;QAQI,WmBpotBO,MAAO,KnBootBG,gBmBpotBH,EnBootBS,KAAM,OmBpotBf,C;QnB  
qotBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAA  
L,EAAV,EAAM,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,kC;MAqGoB,gB;MAHh  
B,gBAAgB,gB;MACHB,WAAW,iBmB/utBJ,MAAO,KnB+utBsB,wBA5FzB,KA4FyB,EAAwB,EAAXB,CmB/utBt  
B,EnB+utBmD,SmB/utBnD,CnB+utBH,C;MACX,QAAQ,C;MACQ,OA9FL,KA8FK,W;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhGqB,GAGP,UAAK,UAAL,EAAK,k  
BAAL,SAhGO,EAAGI,OAHGJ,CAGrB,C;;MAhGT,OAKGO,I;K;IA/FX,kC;MA6GoB,gB;MAHhB,gBAAGB,gB;  
MACHB,WAAW,iBmBjwBJ,MAAO,KnBiwtBsB,wBApGzB,KAoGyB,EAAwB,EAAXB,CmBjwBtB,EnBiwtBm  
D,SmBjwBnD,CnBiwtBH,C;MACX,QAAQ,C;MACQ,OAiGL,KAsGK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,y  
B;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxGqB,GAwGP,UAAK,UAAL,EAAK,kBAAL,SAx  
GO,EAwGI,OAxGJ,CAwGrB,C;;MAxGT,OA0GO,I;K;IAvGX,kC;MAqHoB,gB;MAHhB,gBAAGB,gB;MACHB,  
WAAW,iBmBnxtBJ,MAAO,KnBmxtBsB,wBA5GzB,KA4GyB,EAAwB,EAAXB,CmBnxtBtB,EnBmxtBmD,SmBn  
xtBnD,CnBmxtBH,C;MACX,QAAQ,C;MACQ,OA9GL,KA8GK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAC  
Z,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhHqB,GAGHP,UAAK,UAAL,EAAK,kBAAL,SAhHO,EA  
GHI,OAHHJ,CAGHrB,C;;MAhHT,OAKHO,I;K;IA/GX,kC;MA6HoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iB  
mBrytBJ,MAAO,KnBqytBsB,wBApHzB,KAoHyB,EAAwB,EAAXB,CmBrytBtB,EnBqytBmD,SmBrytBnD,CnBq  
ytBH,C;MACX,QAAQ,C;MACQ,OAiHL,KAsHK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAA  
K,SAAT,C;UAAoB,K;QACpB,IAAK,WAxHqB,GAwHP,UAAK,UAAL,EAAK,kBAAL,SAxHO,EAwHI,OAxHJ,  
CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAqIoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmBvztBJ,M  
AAO,KnBuztBsB,wBA5HzB,KA4HyB,EAAwB,EAAXB,CmBvztBtB,EnBuztBmD,SmBvztBnD,CnBuztBH,C;MA  
CX,QAAQ,C;MACQ,OA9HL,KA8HK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;  
UAAoB,K;QACpB,IAAK,WAhIqB,GAGIP,UAAK,UAAL,EAAK,kBAAL,SAhIO,EAAGI,OAHIJ,CAGIrB,C;;MAhI  
T,OAKIO,I;K;IA/HX,kC;MA6IoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmBz0tBJ,MAAO,KnBy0tBsB,wB  
ApIzB,KAoIyB,EAAwB,EAAXB,CmBz0tBtB,EnBy0tBmD,SmBz0tBnD,CnBy0tBH,C;MACX,QAAQ,C;MACQ,  
OAiIL,KAsIK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK  
,WAxIqB,GAwIP,UAAK,UAAL,EAAK,kBAAL,SAxIO,EAwII,OAxIJ,CAwIrB,C;;MAxIT,OA0IO,I;K;IAvIX,kC;  
MAqJoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmB31tBJ,MAAO,KnB21tBsB,wBA5IzB,KA4IyB,EAAwB  
,EAAXB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;MACX,QAAQ,C;MACQ,OA9IL,KA8IK,W;MAAh  
B,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhJqB,GAGJP,UAA  
K,UAAL,EAAK,kBAAL,SAhJO,EAAGI,OAHIJ,CAGJrB,C;;MAhJT,OAKJO,I;K;IA/IX,kC;MA6JoB,gB;MAHhB,g  
BAAGB,gB;MACHB,WAAW,iBmB72tBJ,MAAO,KnB62tBsB,wBApJzB,KAoJyB,EAAwB,EAAXB,CmB72tBtB,  
EnB62tBmD,SmB72tBnD,CnB62tBH,C;MACX,QAAQ,C;MACQ,OAiJL,KAsJK,W;MAAhB,OAAGB,cAAhB,C;  
QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxJqB,GAwJP,UAAK,UAAL,EAAK,kBA  
AL,SAxJO,EAwJI,OAxJJ,CAwJrB,C;;MAxJT,OA0JO,I;K;IAvJX,kC;MAqKoB,gB;MAHhB,gBAAGB,gB;MACHB

,WAAW,iBmB/3tBJ,MAAO,KnB+3tBsB,wBA5JzB,KA4JyB,EAAwB,EAAxB,CmB/3tBtB,EnB+3tBmD,SmB/3tBnD,CnB+3tBH,C;MACX,QAAQ,C;MACQ,OA9JL,KA8JK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAG,SAAT,C;UAAoB,K;QACpB,IAAK,WAhKqB,GAGKP,sBAAK,UAAAL,EAAK,kBAAL,UAhKO,EAGKI,OAhKJ,CAGKrB,C;;MAhKT,OAKKO,I;K;+EA/JX,yB;MAAA,kF;MAAA,gE;MmB5utBA,iB;MnB4utBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB/utBJ,MAAO,KnB+utBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB/utBtB,EnB+utBmD,SmB/utBnD,CnB+utBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmB9vtBA,iB;MnB8vtBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBjwBJ,MAAO,KnBiwtBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBjwBtB,EnBiwtBmD,SmBjwBnD,CnBiwtBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBhxtBA,iB;MnBgxtBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBnxtBJ,MAAO,KnBmxtBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBnxtBtB,EnBmxtBmD,SmBnxtBnD,CnBmxtBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBlytBA,iB;MnBkytBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBrytBJ,MAAO,KnBqytBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBrytBtB,EnBqytBmD,SmBrytBnD,CnBqytBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBpztBA,iB;MnBoztBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBvztBJ,MAAO,KnBuztBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBvztBtB,EnBuztBmD,SmBvztBnD,CnBuztBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBt0tBA,iB;MnBs0tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBz0tBJ,MAAO,KnBy0tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBz0tBtB,EnBy0tBmD,SmBz0tBnD,CnBy0tBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBx1tBA,iB;MnBw1tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB31tBJ,MAAO,KnB21tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmB12tBA,iB;MnB02tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB72tBJ,MAAO,KnB62tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB72tBtB,EnB62tBmD,SmB72tBnD,CnB62tBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmB53tBA,iB;MnB43tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB/3tBJ,MAAO,KnB+3tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB/3tBtB,EnB+3tBmD,SmB/3tBnD,CnB+3tBH,C;QACX,QAAQ,C;QACQ,uB;QAAGB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,sBAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkBA,kC;MAwFI,WmBh+tBO,MAAO,KnBg+tBG,gBmBh+tBH,EnB+4tBH,KAIkFk,OmBh+tBf,C;MnBi+tBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApFqB,GAoFP,UAAK,CAAL,CAPFO,EAAAnB,KAoFqB,CAAM,CAAN,CAPFF,CAoFrB,C;;MApFT,OAsFO,I;K;IANFX,kC;MA8FI,WmBh/tBO,MAAO,KnBg/tBG,gBmBh/tBH,EnBy5tBH,KAuFkB,OmBh/tBf,C;MnBi/tBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1FqB,GA0FP,UAAK,CAAL,CA1FO,EAAAnB,KA0FqB,CAAM,CAAN,CA1FF,CA0FrB,C;;MA1FT,OA4FO,I;K;IAzFX,kC;MAoGI,WmBhguBO,MAAO,KnBgguBG,gBmBhguBH,EnBm6tBH,KA6FkB,OmBhguBf,C;MnBiguBd,WAAW,iBAAa,IAA



b,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAhGqB,GAgGP,UAAK,CAAL,CAhGO,EAAnB,KA  
gGqB,CAAM,CAAN,CAhGF,CAgGrB,C;;MAhGT,OakGO,I;K;IA/FX,kC;MAOGI,WmBhhuBO,MAAO,KnBghu  
BG,gBmBhhuBH,EnB66tBH,KAmGkB,OmBhhuBf,C;MnBihuBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,  
MAAkB,IAAIB,M;QACI,IAAK,WAtGqB,GAsGP,UAAK,CAAL,CAtGO,EAAnB,KAsGqB,CAAM,CAAN,CAtG  
F,CAsGrB,C;;MAtGT,OAwGO,I;K;IArGX,kC;MAgHI,WmBhiuBO,MAAO,KnBgjuBG,gBmBhiuBH,EnBu7tBH,  
KAyGkB,OmBhiuBf,C;MnBiiuBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK  
,WA5GqB,GA4GP,UAAK,CAAL,CA5GO,EAAnB,KA4GqB,CAAM,CAAN,CA5GF,CA4GrB,C;;MA5GT,OA8G  
O,I;K;IA3GX,kC;MAshI,WmBhjuBO,MAAO,KnBgjuBG,gBmBhjuBH,EnBi8tBH,KA+GkB,OmBhjuBf,C;MnBij  
uBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAIHqB,GakHP,UAAK,CA  
AL,CAIHO,EAAnB,KakHqB,CAAM,CAAN,CAIHF,CAkHrB,C;;MAIHT,OAoHO,I;K;IAjHX,kC;MA4HI,WmBh  
kuBO,MAAO,KnBgkuBG,gBmBhkuBH,EnB28tBH,KAqHkB,OmBhkuBf,C;MnBikuBd,WAAW,iBAAa,IAAb,C;  
MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAxHqB,GAwHP,UAAK,CAAL,CAxHO,EAAnB,KAwH  
qB,CAAM,CAAN,CAxHF,CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAkII,WmBhluBO,MAAO,KnBgluBG,  
gBmBhluBH,EnBq9tBH,KA2HkB,OmBhluBf,C;MnBiluBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAk  
B,IAAIB,M;QACI,IAAK,WA9HqB,GA8HP,sBAAK,CAAL,EA9HO,EA8HE,YA9HrB,KA8HqB,CAAM,CAAN,E  
A9HF,CA8HrB,C;;MA9HT,OAgIO,I;K;+EA7HX,yB;MAAA,gE;MmB99tBA,iB;MnB89tBA,8C;QAQI,WmBh+tB  
O,MAAO,KnBg+tBG,gBmBh+tBH,EnBg+tBS,KAAM,OmBh+tBf,C;QnBi+tBd,WAAW,eAAa,IAAb,C;QACX,aA  
AU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAn  
B,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9+tBA,iB;MnB8+tBA,8C;QAQI,WmBh/tB  
O,MAAO,KnBg/tBG,gBmBh/tBH,EnBg/tBS,KAAM,OmBh/tBf,C;QnBi/tBd,WAAW,eAAa,IAAb,C;QACX,aAAU  
,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,C  
AAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9/tBA,iB;MnB8/tBA,8C;QAQI,WmBhguBO,M  
AAO,KnBggguBG,gBmBhguBH,EnBggguBS,KAAM,OmBhguBf,C;QnBiguBd,WAAW,eAAa,IAAb,C;QACX,aAA  
U,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,  
CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9guBA,iB;MnB8guBA,8C;QAQI,WmBhhuB  
O,MAAO,KnBghuBG,gBmBhhuBH,EnBghuBS,KAAM,OmBhhuBf,C;QnBihuBd,WAAW,eAAa,IAAb,C;QACX,a  
AAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAA  
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9huBA,iB;MnB8huBA,8C;QAQI,WmBhi  
uBO,MAAO,KnBgjuBG,gBmBhiuBH,EnBgjuBS,KAAM,OmBhiuBf,C;QnBiiuBd,WAAW,eAAa,IAAb,C;QACX,a  
AAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAA  
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9iuBA,iB;MnB8iuBA,8C;QAQI,WmBhju  
BO,MAAO,KnBgjuBG,gBmBhjuBH,EnBgjuBS,KAAM,OmBhjuBf,C;QnBijuBd,WAAW,eAAa,IAAb,C;QACX,a  
AAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAA  
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9juBA,iB;MnB8juBA,8C;QAQI,WmBhku  
BO,MAAO,KnBgkuBG,gBmBhkuBH,EnBgkuBS,KAAM,OmBhkuBf,C;QnBikuBd,WAAW,eAAa,IAAb,C;QACX  
,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CA  
AnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MmB9kuBA,iB;MnB8kuBA,8C;Q  
AQI,WmBhluBO,MAAO,KnBgluBG,gBmBhluBH,EnBgluBS,KAAM,OmBhluBf,C;QnBiluBd,WAAW,eAAa,IAA  
b,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,EAAV,EAAMB,kBAA  
M,CAAN,EAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,4F;MAQ8D,yB;QAAA,YAA0B,I;MAAM,sB;QA  
AA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;  
QAAA,YAAoC,I;MAGvN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;  
QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,  
CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,S  
AAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP  
,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E  
;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAA  
O,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAA

U,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iB  
AAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAE  
X,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP  
,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,U  
AAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAF  
hB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IA  
AI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI  
,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,  
K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAA  
O,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;  
QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAcI,I;MAGIN,  
Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q  
ACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,  
C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;  
;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAA  
O,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;  
MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC  
,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,S  
AAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SA  
S,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAA  
Q,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MA  
CxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,S  
AAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAA  
A,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAhB,gB;QAAG  
B,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,  
IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gB  
AAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,S  
AAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ0D,yB;QAAA,YAA0B,I;MAAM,s  
B;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAA  
O,yB;QAAA,YAAyC,I;MAGxN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAGB,SAAh  
B,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,Q  
AAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAE  
P,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAA  
O,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ2D,yB;QAAA,YAA0  
B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YA  
A0B,K;MAAO,yB;QAAA,YAA0C,I;MAG1N,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wB  
AAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QA  
CxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CA  
AP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;Q  
AAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;Q  
AAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;  
QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;  
MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAhB,UAAgB,SAAhB,O;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,g  
BAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAA  
O,UAAU,oBAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ  
,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,0F;M  
AQyC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,  
E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACIN,OAAO,kBAAO,sBAAP,EAAwB,SAAXB,E

AAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQ  
kC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;M  
AAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAA  
mC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQmC  
,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MA  
AI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAm  
C,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQiC,y  
B;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,  
yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC5M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,  
MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQkC,yB;  
QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,y  
B;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,M  
AAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQmC,yB;Q  
AAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;  
QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MA  
AnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQoC,yB;QA  
AA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;Q  
AAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MACIN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAn  
C,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQqC,yB;QAAA,  
YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAA  
A,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MACpN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,E  
AA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAG5F,4F;MAQkC,yB;QAAA,YA  
A0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,Y  
AA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA  
2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAI, W;K;IAQxE,4C;MAAA,mB;QAAE,OAAK  
,qBAAL,eAAK,C;O;K;IAL3B,+B;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,4BAAhB,  
C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QA  
Ae,OAAO,W;MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAL3  
B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;  
QAAE,OAAK,wBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCA  
AgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ  
,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAA  
K,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8  
C;MAAA,mB;QAAE,OAAK,2BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;  
MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,4BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IA  
p8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,  
yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACTB,kCAAgB,8BAAhB,C  
;K;IAUgB,4C;MAAA,mB;QAAE,OAAK,qBAAL,eAAK,C;O;K;IAP3B,+B;MAMI,IA9ggBO,qBAAQ,CA8ggBf,C  
;QAAe,OAAO,e;MACTB,kCAAgB,4BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IA  
P3B,iC;MAMI,IAhhgBO,qBAAQ,CAghgBf,C;QAAe,OAAO,e;MACTB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAA  
A,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAlhgBO,qBAAQ,CAkhgBf,C;QAAe,OAAO,e;MA  
CTB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,wBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAph  
gBO,qBAAQ,CAohgBf,C;QAAe,OAAO,e;MACTB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK  
,yBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAthgBO,qBAAQ,CAshgBf,C;QAAe,OAAO,e;MACTB,kCAAgB,8BAAh  
B,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAxhgBO,qBAAQ,CAwhg  
Bf,C;QAAe,OAAO,e;MACTB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,2BAAL,eAAK,C;O;  
K;IAP3B,iC;MAMI,IA1hgBO,qBAAQ,CA0hgBf,C;QAAe,OAAO,e;MACTB,kCAAgB,8BAAhB,C;K;IAUgB,8C;  
MAAA,mB;QAAE,OAAK,4BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IA5hgBO,qBAAQ,CA4hgBf,C;QAAe,OAAO,



CI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAA S,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;K;sGA GX,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcS,sBAAI,KAAJ,CAAtC, GAAsD,aAAa,KAAb,C;O;KAPjE,C;IAUA,6C;MACqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OAAy,YAAL,SAAK,E AAU,KAAV,C;MACHb,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO, QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;; MAEf,OAAO,I;K;sGAGX,yB;MAAA,sD;MAAA,mC;QAOI,OAAy,UAAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C; gFAUA,gC;MAOW,sB;;QAyHS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAzHH,SAyHO,C AAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA1HP,yB;K;wFAGJ,gC;MA6VoB,Q ;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAvVc,SAuVV,CAAU,OAAV,CAA J,C;UACI,OAAO,O;;;MAxVf,OA2VO,I;K;wFAxVX,gC;MAOW,qB;;QA0VP,eAAoB,+BAAa,cAAb,C;QACpB,O AAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IA7Vc,SA6VV,CAAU,OAAV,CAAJ,C;YAAwB,oBAAO, O;YAAP,sB;;;QAE5B,oBAAO,I;;;MA/VP,wB;K;IAGJ,6B;MAOQ,kBADE,SACF,Q;QAAW,OAAy,SAAL,SAAK ,C;;QAE5B,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,OAAO,QA AS,O;;K;IAK5B,6B;MAMI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,CAAL, C;K;mFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA AI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,wDAAvB,C;O;KANV,C;oGASA,yB;M AAA,iE;MAAA,uC;QAS8C,IAAnC,I;QAAA,+B;;UAYS,U;UAAA,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B; YACZ,aAbwB,SAaX,CAAU,OAAV,C;YAcB,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAI BA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,mEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q; MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,O AAO,M;;;MAGf,OAAO,I;K;IAGX,mC;MAKQ,kBADE,SACF,Q;QACI,IAAI,mBAAJ,C;UACI,OAAO,I;;UAEP,O AAO,sBAAK,CAAL,C;;QAGX,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,OAAO,QA AS,O;;K;IAK5B,mC;MAII,OAAW,mBAAJ,GAAe,IAAf,GAAYB,sBAAK,CAAL,C;K;+FAGpC,gC;MAIoB,Q;MA AA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACr D,OAAO,I;K;0FAGX,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcS,sBA AI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BA A3B,GAAcS,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMiB,Q;MAFb,IAAI,8BAAJ,C;QAAkB,OAAO,SA AK,eAAQ,OAAR,C;MAC9B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C ;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;IAGX,uC;MAKI,OAAO,w BAAQ,OAAR,C;K;gGAGX,yB;MAAA,wE;MAAA,uC;QAKiB,Q;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAA b,C;UAAa,sB;UACT,mBAAmB,KAAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAE J,OAAO,E;O;KAXX,C;gGAcA,gC;MAKiB,Q;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAC T,IAAI,UAAU,IAAV,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;8FAGX,yB;MAAA,wE;MAAA,uC ;QAMiB,Q;QAFb,gBAAgB,E;QACHb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,K AAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAY,K;UACHb,qB;;QAEJ,OAAO,S;O;KAZX,C;8FAeA,g C;MAII,eAAe,SAAK,sBAaA,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,IAAI,UAAU,QAAS,WAAAnB,CA AJ,C;UACI,OAAO,QAAS,Y;;MAGxB,OAAO,E;K;IAGX,4B;MASQ,kBADE,SACF,Q;QAAW,OAAy,QAAL,S AAK,C;;QAE5B,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,WAA W,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,4B;MAQI,IA AI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,2BAAL,C;K;iFAGX,yB;MAAA,iE;M AAA,gB;MAAA,8B;MAAA,uC;QAUoB,UAQT,M;QAVP,WAAe,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,c AAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAA C,KAAL,C;UAAy,MAAM,gCAAuB,wDAAvB,C;QAEIB,OAAO,2E;O;KAlBX,C;iFAqBA,yB;MAAA,iE;MAAA, uC;QAQI,eAAe,SAAK,sBAaA,cAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IAAI,U AAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAE5C,MAAM,gCAAuB,kDAAvB,C;O;KAbV,C;IAGBA,2C;MAOiB, Q;MAHb,IAAI,8BAAJ,C;QAAkB,OAAO,SAAK,mBAAy,OAAZ,C;MAC9B,gBAAgB,E;MACHb,YAAY,C;MA CC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UAC

I,YAAY,K;QACHB,qB;;MAEJ,OAAO,S;K;IAGX,2C;MAKI,OAAO,4BAAY,OAAZ,C;K;IAGX,kC;MAOQ,kBAD  
E,SACF,Q;QAAW,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;;QAEvC,eAAe,oB;QA  
Cf,IAAI,CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,WAAW,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UA  
CI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,kC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAA  
O,CAAP,IAAL,C;K;6FAGpC,gC;MAOoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y  
B;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO,O;;MAGf,OAAO,I;K;6FAGX,gC;MAMI,eAAe,SAAK,sBA  
Aa,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,cAAc,QAAS,W;QACvB,IAAI,UAAU,OAAV,CAAJ,C;UAA  
wB,OAAO,O;;MAEnC,OAAO,I;K;qFAGX,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C  
;O;KARX,C;IAWA,sC;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,sBAAvB,C;MACV,OAAO,qBAAU,MAA  
O,iBAAQ,cAAR,CAAJB,C;K;iGAGX,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;K  
APX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAJB,  
C;K;IAGX,8B;MAKQ,kBADE,SACF,Q;QAAW,OAAy,UAAAL,SAAK,C;;QAEEnB,eAAe,oB;QACf,IAAI,CAAC,Q  
AAS,UAAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,  
MAAM,gCAAYB,uCAAZB,C;QACV,OAAO,M;;K;IAKnB,8B;MALiB,IAAN,I;MAAA,QAAM,cAAN,C;aACH,C;  
UAAK,MAAM,2BAAuB,gBAAvB,C;aACX,C;UAAK,6BAAK,CAAL,C;UAAAL,K;;UACQ,MAAM,gCAAYB,iCA  
AzB,C;;MAHIB,W;K;qFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QA  
XP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CA  
AJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,qDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I;;QAGhB,IAAI  
,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,wDAAvB,C;QAEIB,OAAO,6E;O;KafX,C;IAkBA,oC;MAKQ,kBADE  
,SACF,Q;QAAW,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;;QAE1C,eAAe,oB;QACf,IAAI,  
CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,OAAO,I;QA  
CX,OAAO,M;;K;IAKnB,oC;MAII,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;iGAGvC,g  
C;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI  
,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;MAGhB,IAAI,  
CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAoBsC,UAGT,MAHS,EAARB,M;MN/pBb,IA  
AI,EMsoBI,KAAK,CNtoBT,CAAJ,C;QACI,cMqoBc,sD;QNpoBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMqoBV  
,IAAI,MAAK,CAAT,C;QAAy,OAAO,mB;MACnB,Q;MACA,IAAI,oCAAJ,C;QACI,iBAAiB,iBAAO,CAAP,I;Q  
ACjB,IAAI,cAAc,CAAIb,C;UACI,OAAO,W;QACX,IAAI,eAAc,CAAIb,C;UACI,OAAO,OAAO,kBAAP,C;QAC  
X,OAAO,iBAAa,UAAb,C;QACP,IAAI,8BAAJ,C;UACI,IAAI,sCAAJ,C;YAC0B,qB;YAAtB,iBAAc,CAAd,wB;cA  
CI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;YAEI,wCAAA,CAAb,C;YAAb,OAAa,gBAAb,C;cAAa,wB;cACT,IAA  
K,WAAI,IAAJ,C;;UAEB,OAAO,I;;QAIX,OAAO,gB;;MAEX,YAAY,C;MACC,6B;MAAb,OAAa,gBAAb,C;QA  
Aa,0B;QACT,IAAI,SAAS,CAAb,C;UAAgB,IAAK,WAAI,MAAJ,C;;UAAe,qB;;MAEXc,OAAy,qBAAL,IAAK,C  
;K;IAGhB,kC;MNRqBI,IAAI,EM6qBI,KAAK,CN7qBT,CAAJ,C;QACI,cM4qBc,sD;QN3qBd,MAAM,gCAAYB,O  
AAQ,WAAjC,C;;MM4qBV,OAAO,kBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;kGAGX,y  
B;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,IAAI,CAAC,mBAAL,C;UACI,eAAe,+BAAa,cAAb,C;UACf,OAAO,  
QAAS,cAAhB,C;YACI,IAAI,CAAC,UAAU,QAAS,WAAAnB,CAAL,C;cACI,OAAO,gBAAK,QAAS,YAAT,GAA  
uB,CAAvB,IAAL,C;;QAIInB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,  
K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,  
C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;QAEEnB,OAAO,I;O;  
KafX,C;oFAkBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA2FA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,  
C;UAAgB,yB;UAAM,IA3FU,SA2FN,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QA3FID,OA4F  
O,W;O;KAIGX,C;kGASA,yB;MAAA,+D;MAiKCA,wE;MAjKCA,uC;QAQW,kBAAgB,gB;QAGkCV,gB;QADb,Y  
AAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UApjCT,IAZmC,SAY/B,CAojCkB,oBAAMb,cAANb,EAAM  
B,sBAANb,UApjCIB,EAojC+C,IApjC/C,CAAJ,C;YAA2C,sBAojCQ,IApjCR,C;;QAZ/C,OAco,W;O;KATBX,C;S  
AWA,yB;MASjCA,wE;MATjCA,oD;QA6jCiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;U  
ApjCT,IAAI,UAOjCkB,oBAAMb,cAANb,EAAMb,sBAANb,UApjCIB,EAojC+C,IApjC/C,CAAJ,C;YAA2C,sBAo  
jCQ,IApjCR,C;;QAE/C,OAAO,W;O;KAXX,C;wGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAMb,gB;QASV,Q  
;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,YAAJ,C;YAAkB,WAAy,WAAI,OAAJ,C;;QAT

pD,OAUO,W;O;KAhBX,C;4GASA,4C;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,YAAJ,C;UAAKB,WAAY,WAAI,OAAJ,C;;MACpD,OAAO,W;K;0FAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA4BH,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CA5BS,SA4BR,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QA5B3D,OA6BO,W;O;KAnCX,C;IASA,oC;MAMI,OAAO,6BAAgB,gBAAhB,C;K;IAGX,mD;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAY,WAAI,OAAJ,C;;MACvD,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OhB7wBe,W;;MgB8wBtC,OAA6D,SAAtD,SAAK,iBAAQ,OAAQ,MAAhB,EAAuB,OAAQ,aAAR,GAAuB,CAAvB,IAAvB,CAAI,D,C;K;IAGjE,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAAa,IAAb,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAL,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,8B;MAGBiB,Q;MN91Bb,IAAI,EMs1BI,KAAK,CNt1BT,CAAJ,C;QACI,cMq1Bc,sD;QNp1Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMq1BV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,oCAAJ,C;QACI,IAAI,KAAK,cAAT,C;UAAe,OAAO,mB;QACTb,IAAI,MAAK,CAAT,C;UAAy,OAAO,OAAO,mBAAP,C;;MAEvB,YAAy,C;MACZ,WAAW,iBAAa,CAAb,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAy,qBAAL,IAAK,C;K;IAGhB,kC;MAeqC,IAGhB,I;MNx3BjB,IAAI,EM82BI,KAAK,CN92BT,CAAJ,C;QACI,cM62Bc,sD;QN52Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MM62BV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,kBAAP,C;MACnB,WAAW,iBAAa,CAAb,C;MACX,IAAI,sCAAJ,C;QACI,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;UACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;QAEI,sCAaa,OAAO,CAAP,IAAb,C;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAK,WAAI,IAAJ,C;;MAEb,OAAO,I;K;kGAGX,yB;MAAA,qD;MAAA,gE;MAAA,gD;MAAA,uC;QAMI,IAAI,mBAAJ,C;UACI,OAAO,W;QACX,eAAe,+BAAa,cAAb,C;QACf,OA AO,QAAS,cAAhB,C;UACI,IAAI,CAAC,UAAU,QAAS,WAAhB,CAAL,C;YACI,QAAS,O;YACT,mBAAmB,iBA AO,QAAS,YAAhB,I;YACnB,IAAI,iBAAGB,CAApB,C;cAAuB,OAAO,W;YACI,kBAA3B,eAAa,YAAb,C;YACH ,OAAGB,kBAAhB,C;cACI,sBAaa,eAAb,C;YAFR,OH51BD,W;;QGk2BP,OAAO,iB;O;KApBX,C;0FAuBA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CA AC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAoBA,+B;MAII,IAAI,wCAAsB,kBAAQ,CAAIC,C;QAAqC,OAAO,mB;MAC5C,WAAW,0B;MACN,WAAW,IAAK,C;MACL,OAA O,I;K;IAGX,uC;MAOI,aAAU,2BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;Q ACf,sBAAK,CAAL,EAAU,SAAK,aAAI,CAAJ,EAAO,sBAAK,CAAL,CAAP,CAAf,C;;K;oFAIR,yB;MAAA,oD; MJr4BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,c AAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MI83Bf,sC;QAMI,IAAI,iBA AO,CAAX,C;UAAc,oBJp4Bd,eAAW,iBIo4BsB,QJp4BtB,CAAX,CiO4Bc,C;;O;KANIB,C;wGASA,yB;MAAA,oD; MJ33BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,c AAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MIo3Bf,sC;QAMI,IAAI,iBA AO,CAAX,C;UAAc,oBJ13Bd,eAAW,2BI03BgC,QJ13BhC,CAAX,CiO3Bc,C;;O;KANIB,C;IASA,sC;MAMI,sBAAS,c AAT,C;K;IAGJ,6B;MASgB,Q;MAHZ,IAAI,oCAAJ,C;QACI,IAAI,kBAAQ,CAAZ,C;UAAe,OAAy,SAAL,SAAK ,C;QAEwB,kBAA3C,sBC7+Bsd,sBD6+BtD,uB;QAAmD,mB;QAA3D,OAAoE,OHp7BjE,WGo7BiE,C;;MAEjD,k BAAhB,0B;MAAwB,oB;MAA/B,OHt7BO,W;K;wFGy7BX,yB;MAAA,wD;MJ96BA,sC;MAAA,oC;MAAA,uBA Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA 2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MIu6Bf,sC;QAQI,OAAO,sBJ/6BP,eAAW,iBI+6BiB,QJ/6BjB,CA AX,Ci+6BO,C;O;KARX,C;4GAWA,yB;MAAA,wD;MJt6BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe ,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3 B,C;W;S;OA+EI,C;MI+5Bf,sC;QAMI,OAAO,sBJr6BP,eAAW,2BIq6B2B,QJr6B3B,CAAX,CiQ6BO,C;O;KANX, C;IASA,uC;MAMI,OAAO,wBAAW,cAAX,C;K;IAGX,6C;MASE,Q;MAHX,IAAI,oCAAJ,C;QACG,IAAI,kBAAQ ,CAAZ,C;UAAe,OAAy,SAAL,SAAK,C;QAEe,kBAAIC,sBCxhCuD,sBDwhCvD,uB;QAA0C,iC;QAAID,OAAyE,

OH/9BrE,WG+9BqE,C;;MAErD,kBAAhB,0B;MAAwB,mC;MAA/B,OHj+BO,W;K;IGo+BX,qC;MAMoB,UACL,  
M;MAHX,aAAa,oBAAa,cAAb,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OA  
AO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,cAAU,cA  
AV,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,Y  
AAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAY,C;M  
ACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,oC;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,  
M;K;IAGX,oC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAZ,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAGB,c  
AAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,mC;MAMoB,U  
ACL,M;MAHX,aAAa,iBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QA  
CZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,iC;MAMoB,UACL,M;MAHX,aAAa,eA  
AS,cAAT,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sB  
AAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAA  
Y,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,  
OAAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,eAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,O  
AAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;0FAGX,yB;M  
AAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,  
CAAYC,EAAC,EAAD,C;QACjD,kBAAY,mBAAoB,QAAPB,C;QAYEH,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;  
UAAgB,yB;UACZ,WA1E8C,SA0E/B,CAAU,OAAV,C;UflkBnB,wBAAl,IAAK,MAAT,EAAGB,IAAK,OAArB,C  
;;QewfA,OA4EO,W;O;KAXFX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eA  
AwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAD,C;QACjD,kBAAC,mBAAoB,QAAPB,C;QA  
2BL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA5BoC,WA4BhC,CAAY,OAAZ,CAA  
J,EA0B,OAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE  
;MAAA,yD;QAUl,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAD,C;QACjD,kBAAC,m  
BAAoB,QAAPB,C;QA6BL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA9BoC,WA8B  
hC,CAAY,OAAZ,CAAJ,EA9BiD,cA8BvB,CAAe,OAAf,CAA1B,C;;QA9BhB,OA9CO,W;O;KA3CX,C;mGAcA,+  
C;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EA  
A0B,OAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,y  
B;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;8FAGX,6C;  
MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAe,UAAU,OAAV,C;QflkBnB,wBAAl,  
IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MeokBA,OAAO,W;K;kGAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;M  
AAA,uE;MAAA,2C;QAYl,aAAa,mBAA6D,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAD,CAA  
7D,C;QAcG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAbO,MAcP,aAAI,OAAJ,EAd,eAcF,CAAc,  
OAAc,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;sGAgBA,iD;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;  
QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EA0B,cAAc,OAAc,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;  
MAAA,2B;MAAb,OAAa,cAAb,C;QAAs,sB;QACT,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAI  
l,OAAO,0BAAa,eAAW,YAAY,mCAAwB,EAAXB,CAAZ,CAAX,CAAb,C;K;IAGX,6B;MAKqB,IAAN,I;MADX,  
IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,kB;YAAL,K;eACA,C;YAAK,cAAW,8BAAJ,GAAkB,s  
BAAl,CAAJ,CAAIB,GAA8B,oBAAW,OAAhD,C;YAAL,K;;YACa,uBAAL,SAAK,C;YAHV,K;;QAAP,W;;MAM  
J,OAA4B,qBAAhB,gBAAL,SAAK,CAAGB,C;K;IAGhC,oC;MAII,IAAI,oCAAJ,C;QACI,OAAy,gBAAL,SAAK,  
C;MACHB,OAAO,0BAAa,gBAAb,C;K;IAGX,oC;MAII,OAAO,iBAAU,SAAV,C;K;IAGX,4B;MAOqB,IAAN,I;M  
ADX,IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAA  
kB,sBAAK,CAAL,CAAIB,GAA+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,iCAAa,qBAAiB,YAAY,cAAZ,CAAjB,  
CAAb,C;YAhL,K;;QAAP,W;;MAMJ,OAAwC,oBAAjC,0BAAa,sBAAb,CAAiC,C;K;sFAG5C,yB;MAAA,+D;M  
AwFA,gD;MAxFA,uC;QAMW,kBAAU,gB;QAsFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,  
WAvF6B,SAuFIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxFhB,OA0FO,W;O;KAhGX,C;uF  
ASA,yB;MAAA,+D;MA0FA,gD;MA1FA,uC;QAUW,kBAAU,gB;QAwFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB  
,C;UAAgB,yB;UACZ,WAZF6B,SAYFIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1FhB,OA4F  
O,W;O;KAtGX,C;oGAaA,yB;MAAA,+D;MA8BA,wE;MAAA,gD;MA9BA,uC;QAYW,kBAAiB,gB;QA6BR,gB;



QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA9BoC,SA8BzB,CAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA/BhB,OAiCO,W;O;KA7CX,C;oGAeA,yB;MAAA,+D;MAiCA,wE;MAAA,gD;MAjCA,uC;QAYW,kBAAiB,gB;QAgCR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WajCoC,SAiCzB,CAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAIChB,OAoCO,W;O;KAhDX,C;wGAeA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;uFAeA,yB;MAAA,wE;MAyBA,+D;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA1BiD,WA0BvC,CAAY,OAAZ,C;UfrnCP,U;UADP,YeunCe,WfvnCH,WeunCwB,GfvcxB,C;UACL,IAAI,aAAJ,C;YACH,aeqnCuC,gB;YAA5B,WfpcCX,aeonCgC,GfpcChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UeinCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA5BT,OA8BO,W;O;KAvcX,C;uFAYA,yB;MAAA,wE;MA8BA,+D;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA/BiD,WA+BvC,CAAY,OAAZ,C;UfvoCP,U;UADP,YeyoCe,WfzoCH,WeyoCwB,GfzoCxB,C;UACL,IAAI,aAAJ,C;YACH,aeuoCuC,gB;YAA5B,WftoCX,aeoCgC,GftoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UemoCA,iB;UACA,IAAK,WAAI,eA AjCyD,cAiCrD,CAAE,OAAf,CAAJ,C;;QAJCT,OAmCO,W;O;KA7CX,C;0FAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UfrnCP,U;UADP,YeunCe,WfvnCH,WeunCwB,GfvcxB,C;UACL,IAAI,aAAJ,C;YACH,aeqnCuC,gB;YAA5B,WfpcCX,aeonCgC,GfpcChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UeinCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UfvoCP,U;UADP,YeyoCe,WfzoCH,WeyoCwB,GfzoCxB,C;UACL,IAAI,aAAJ,C;YACH,aeuoCuC,gB;YAA5B,WftoCX,aeoCgC,GftoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UemoCA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;4FAkBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,Y;QAA6C,OAAA,oBAAgB,W;O;MATrE,iDAUQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;8EAeA,yB;MAAA,kF;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,mCAAwB,EAAXB,CAAb,C;QAUeA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAXeWc,SAwEpC,CAAU,IAAV,CAAJ,C;;QAxehB,OAYEO,W;O;KAhFX,C;4FAUA,yB;MAAA,kF;MAAA,gE;MA+BA,wE;MA/BA,uC;QAOW,kBAAa,eAAa,mCAAwB,EAAXB,CAAb,C;QAGCP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WajC+C,SAiC3C,CAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,IAAvC,CAAJ,C;;QAJChB,OAkCO,W;O;KAZCX,C;0GAUA,yB;MAAA,+D;MAwSA,wE;MAxSA,uC;QAOW,kBAAoB,gB;QAwSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA9RsB,U;UAAA,cAVQ,SAUR,CA8RT,oBAAmB,cAAAnB,EAAMb,sBAAnB,UA9RS,EA8RoB,IA9RpB,W;YAA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;8GAUA,yB;MA8RA,wE;MA9RA,oD;QAqSiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA9RsB,U;UAAA,wBA8RT,oBAAmB,cAAAnB,EAAMb,sBAAnB,UA9RS,EA8RoB,IA9RpB,W;YAA6C,6B;;;QACHf,OAAO,W;O;KARX,C;+FAWA,yB;MAAA,wE;MAAA,oD;QAQiB,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAAI,UAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAuC,IAAvC,CAAJ,C;;QACHb,OAAO,W;O;KAVX,C;4FAaA,yB;MAAA,+D;MAAA,uC;QAOW,kBAaAa,gB;QA4PJ,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UApPK,U;UAAA,cARe,SAQf,CAoPQ,OApPR,W;YAAc,6B;;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAYPoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UApPK,U;UAAA,wBAoPQ,OApPR,W;YAAc,6B;;;QAC3D,OAAO,W;O;KANX,C;kFASA,6C;MAKiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;IAQiB,4C;MAAA,mB;QAAE,gC;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAJb,C;K;IAGX

,+B;MASI,OAA2B,SAAf,eAAL,SAAK,CAAE,C;K;4FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACD,2B;QAAV,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAL,WAAI,GAAL,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;IAoBA,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;MAMiB,IAAN,I;MACH,kBADS,SACT,c;QAAoB,4BAAc,SAAd,C;;QACZ,iCAAs,sBAAb,C;MAFZ,W;K;IAMJ,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAL,EAAO,KAAP,C;MACJ,OAAO,G;K;8EAGX,yB;MAAA,gD;MAAA,uC;QAWoB,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAZX,C;IAeA,2B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,CAAC,mB;MACHc,OAAO,oBAAW,U;K;+EAGtB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,K;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KARX,C;IAWA,6B;MAMoB,Q;MAFhB,IAAI,oCAAJ,C;QAAwB,OAAO,c;MAC/B,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MACtB,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,yB;MAAA,gD;MAAA,wE;MAAA,uC;QAMoB,Q;QAFhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,C;QAC5C,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;QAC9C,OAAO,K;O;KAPX,C;gFAUA,yC;MAUoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,wE;MAAA,gD;QAYoB,UAAiD,M;QAFjE,YAAy,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAAnB,EAAmB,sBAAnB,UAAV,EAAuC,WAAvC,EAAoD,OAApD,C;;QACpC,OAAO,W;O;KAbX,C;0FAgBA,yC;MASI,kBAAkB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;MAGtB,OAAO,W;K;wGAGX,yC;MAUI,kBAAkB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,YAAy,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAAS,WAA1B,EAAc,WAAtC,C;;MAGtB,OAAO,W;K;sFAGX,6B;MAKoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;oGAG1B,yB;MAAA,wE;MAAA,oC;QAOiB,UAAgC,M;QAD7C,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAAnB,EAAmB,sBAAnB,UAAP,EA AoC,IAAP,C,C;;O;KAPvB,C;IAUA,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF5wDG,MAAO,KE4wDE,GF5wDF,EE4wDO,CF5wDP,C;;ME8wDd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF5yDG,MAAO,KE4yDE,GF5yDF,EE4yDO,CF5yDP,C;;ME8yDd,OAAO,G;K;IAGX,2B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KAXBX,C;8FA2BA,+B;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,OAAO,O;K;mFAGX,yB;MAAA,sE;MFn3DA,iB;ME3DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF73DG,MAAO,KE63DO,QF73DP,EE63DiB,CF73DjB,C;;QE+3Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MFr5DA,iB;MEq5DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF/5DG,MAAO,KE+5DO,QF/5DP,EE+5DiB,CF/5DjB,C;;QEi6Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe

,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2B  
AAW,CAAX,KA AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFx7DA,iB;MEw7DA,sC  
;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;Q  
ACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WfH8DG,MAAO,KEg8DO,QFh8DP  
,EEg8DiB,CFh8DjB,C;;QEK8Dd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MFx9DA,iB;MEw9DA,sC;QAWI,eAAe,oB;  
QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAA  
S,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WfH+DG,MAAO,KEg+DO,QFh+DP,EEg+DiB,CFh+  
DjB,C;;QEK+Dd,OAAO,Q;O;KAIBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAy  
B,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,  
OAAIB,C;QACR,IAAI,2BAAW,CAAX,KA AJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MAAA,sE;  
MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS  
,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,Q  
AAR,EAakB,CAAI,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAuBA,2C;  
MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAIB,C;M  
ACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAakB  
,CAAI,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MACf,IA  
AI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,  
QAAQ,QAAS,O;QACjB,MFviEG,MAAO,KEuiEE,GFviEF,EEuiEO,CFviEP,C;;MEyiEd,OAAO,G;K;IAGX,iC;M  
AOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,  
QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFvnkEG,MAAO,KEmkEE,GFnkEF,EEmkEO,CFnkEP,C;;MEq  
kEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAA  
U,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KA AJ,C;UA  
Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MA  
AM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAA  
W,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MA  
KI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,Q  
AAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAj  
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q  
AAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,M  
F16DG,MAAO,KE06DE,GF16DF,EE06DO,CF16DP,C;;ME46Dd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,  
IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;Q  
ACI,QAAQ,QAAS,O;QACjB,MF18DG,MAAO,KE08DE,GF18DF,EE08DO,CF18DP,C;;ME48Dd,OAAO,G;K;IA  
GX,2B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MAC  
nB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KA AJ,C;UAAa,MAAM,C;;M  
AEvB,OAAO,G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAy  
B,MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SA  
AS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA AJ,C;YA  
CI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KAXBX,C;8FA2BA,+B;MAOI,eA  
Ae,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAA  
S,UAAAd,C;QAAyB,OAAO,O;MAChC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CA  
AT,C;QACR,IAAI,2BAAW,CAAX,KA AJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACI  
B,OAAO,O;K;mFAGX,yB;MAAA,sE;MFjhEA,iB;MEihEA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,  
C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAA  
S,QAAS,OAAIB,C;UACR,Wf3hEG,MAAO,KE2hEO,QF3hEP,EE2hEiB,CF3hEjB,C;;QE6hEd,OAAO,Q;O;KAp  
BX,C;mFAuBA,yB;MAAA,sE;MFnjEA,iB;MEMjEA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UA  
AyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QA  
AS,OAAIB,C;UACR,Wf7jEG,MAAO,KE6jEO,QF7jEP,EE6jEiB,CF7jEjB,C;;QE+jEd,OAAO,Q;O;KApBX,C;mF  
AuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/

B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFtIEA,iB;MEsIEA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF9IEG,MAAO,KE8IEO,QF9IEP,EE8IEiB,CF9IEjB,C;;QEgmEd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MFtnEA,iB;MEsnEA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF9nEG,MAAO,KE8nEO,QF9nEP,EE8nEiB,CF9nEjB,C;;QEgoEd,OAAO,Q;O;KAlBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MAAA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAuBA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFrS EG,MAAO,KEqsEE,GFrsEF,EEqsEO,CFrsEP,C;;MEusEd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFjuEG,MAAO,KEiuEE,GFjuEF,EEiuEO,CFjuEP,C;;MEMuEd,OA AO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,G AAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAKI,eAAe,o B;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAh B,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC ,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,mB;MAC/B,OAAO,CAAC,o BAAW,U;K;iFAGvB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO, I;QAC5B,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,K;;Q ACrD,OAAO,I;O;KARX,C;oFAWA,6B;MAKmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM, OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MA3wBA,wE;MA2wBA,2BAQIB,yB;QAnxBj B,wE;eAmxBiB,0B;UAAA,4B;YAAE,aAAe,c;YA5wBjB,gB;YADb,YAAY,C;YACC,2B;YAAb,OAAa,cAAb,C;c AAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAMB,sBAAnB,UAAP,EAAoC,IAApC,C;;YA4wBmB,W;W;S;OAAz B,C;MARjB,oC;QApwBiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBA AmB,cAAnB,EAAMB,sBAAnB,UAAP,EAAoC,IAApC,C;;QA4wBnB,gB;O;KARJ,C;oFAWA,yB;MAAA,4F;MA AA,uC;QAaI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC /B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAauB,QAAS,OAAhC,C;; QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAKBmD,Q;QAL/C,eAAe,SAAK ,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,YAAY,C;QACZ,kBAA qB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMB,oBAAnB,QA AV,EAauC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;8GAuBA,yB;MAAA,wE;MAAA,uC ;QAKBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,YAAY,C; QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMB,oB AAnB,QA AV,EAauC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,gC;MAcI,e AAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,kBAAqB,QAAS,O;MAC9B,OA AO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAauB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;8FAGX,yB; MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,

8BAA9B,C;QACV,kBAaQb,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAhB,EA  
A+B,WAA/B,C;;QAEIB,OAAO,W;O;KArBX,C;4GAuBA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C  
;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAaQb,QAAS,W;QAC9B,OA  
AO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAc,WA  
AtC,C;;QAEIB,OAAO,W;O;KArBX,C;wHAWBA,gC;MAaI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cA  
Ad,C;QACI,OAAO,I;MACX,kBAaQb,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,YAAY,QAAS,gB;QAC  
rB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAc,WAAtC,C;;MAEIB,OAAO,W;K;0GAGX,gC;MAcI,eAA  
e,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAaQb,QAAS,W;MAC9B,OAA  
O,QAAS,cAAhB,C;QACI,cAAc,UAAU,QAAS,WAAhB,EAA+B,WAA/B,C;;MAEIB,OAAO,W;K;8FAGX,yB;M  
AAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAJhB,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kB  
AAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;  
QAArD,aH7sFO,W;QG8sFP,kBAAkB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,  
WAAV,EAAuB,OAavB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,  
kF;MAAA,gD;MAAA,gE;MAAA,gD;QAmBoB,UACY,M;QAN5B,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,  
kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,  
8B;QAArD,aHtuFO,W;QGuuFP,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;  
UACZ,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAhB,EAAGC,OAAhC,C;UACd,MAAO,WAAI,WAAJ,  
C;;QAEX,OAAO,M;O;KAvBX,C;kGA0BA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAcI,eAAe,SAAK,  
W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACc,sBAaQb,QAAS,OAA9B,C;QACuD,kBAA1  
C,eAAa,mCAAwB,EAAXB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aHjwFO,W;QGkwFP,OAAO,QAAS,UAAh  
B,C;UACI,gBAAc,UAAU,aAAV,EAAuB,QAAS,OAAhC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;K  
AtBX,C;gHAyBA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAoBgC,Q;QAN5B,eAAe,SAAK,W;QACpB  
,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACc,sBAaQb,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,m  
CAAwB,EAAXB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aH1xFO,W;QG2xFP,YAAY,C;QACZ,OAAO,QAAS,  
UAAhB,C;UACI,gBAAc,WAAU,YAAV,EAAU,oBAAV,SAAmB,aAAnB,EAAGC,QAAS,OAAzC,C;UACd,MAA  
O,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;gFA0BA,yB;MArGA,kF;MAAA,gD;MAAA,gE;MAqGA,gD;Q  
AcW,sB;;UAIGS,Q;UAJhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,qBAAO,O  
AqGZ,OArGY,C;YAAP,uB;;UACqB,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBAoGIC,OApgkC,C;  
UAArD,aH7sFO,W;UG8sFP,kBAAmGmB,O;UAIGH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAiGwB,S  
AjGV,CAAU,WAAV,EAAuB,OAavB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;QA8FP,yB;O;KAd  
J,C;8FAiBA,yB;MA9FA,kF;MAAA,gD;MAAA,gE;MA8FA,gD;QAEW,6B;;UA1FS,gB;UALhB,oBAAoB,mCAA  
wB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,4BAAO,OA8FL,OA9FK,C;YAAP,8B;;UACqB,kBAAh  
C,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBA6F3B,OA7F2B,C;UAArD,aHtuFO,W;UGuuFP,YAAY,C;UACZ,  
kBA2F0B,O;UA1FV,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAyF+B,SAzFjB,EAAU,cAAV,EAAU,s  
BAAV,WAAmB,WAAhB,EAAGC,OAAhC,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAsFP,gC;O;  
KAFJ,C;kFakBA,+B;MAOoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,YA  
AO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,O  
AAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADh  
B,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G  
;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,YAAO,S  
AAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QAC  
gB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C  
;mFAGBA,yB;MIBjIFA,6B;MkBiIFA,sC;QAWoB,Q;QADhB,UIBjIFmC,ckBilFnB,CIBjIFmB,C;QkKlFnB,2B;QA  
AhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MIB/5FiD,ckB+5FjD,GIB/5F2D,KAAK,GkB+5FzD,SAAS,OAAT,Cl  
B/5FoE,KAAx,IAAf,C;;QkBi6FrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MD/IFA,+B;MC+IFA,sC;QAWoB,Q;QAD  
hB,UD9IFqC,eAAW,oBC8IF/B,CD9IF+B,CAAX,C;QC+IFrB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,  
MD76FmD,eC66FnD,GD76F8D,KAAK,KC66F5D,SAAS,OAAT,CD76FuE,KAAx,CAAhB,C;;QC+6FvD,OAAO,  
G;O;KAdX,C;IAiBA,qC;MAIoB,UAMT,M;MANS,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,IAAI,eAA

J,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB,MAAzB,C;;MAId,OAAO,mE;K;IAGX,qC;MAIoB,UAMT,M;MA NS,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,eAAJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB, MAAzB,C;;MAId,OAAO,+D;K;IAGX,kC;MAWI,OAAO,oBAAS,IAAT,EAAe,IAAf,EAAc,IAAT,C,C;K;IAGX, +C;MAGBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAT,C,EAAwD,SAAXD,C;K;IAGX,mC;MAII,aAAa,iBAA a,mCAAwB,EAAxB,CAAb,C;MACb,kBAAC,KAAAd,C;MA7uEgB,Q;MAAA,OA8uET,SA9uES,W;MAAhB,OAAG B,cAAhB,C;QAAGB,2B;QAAU,oB;QA8uEK,IAAI,CAAC,SAAD,IAAY,OA9uEX,SA8uEW,UAAhB,C;UAAiC, YAAU,I;UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QA9uEvE,qB;UA8uED,MA9uEqC,WAAI,SAAJ,C;;MA8uE1 D,OAAqB,M;K;IAGzB,sC;MAII,IAAI,QrBypJG,YAAQ,CqBzpJf,C;QAAwB,OAAy,SAAL,SAAK,C;MA3xE7B, kBAAy,gB;MA4BH,Q;MAAA,OAgwET,SAhwES,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAg wEF,mBAhwEa,OAgwEb,CAhwEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAGwE3D,OA/vEO,W;K;IAkwEX,sC; MAII,YAAqB,6BAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAy,SAAL,SAAK,C;MAryET,kBAAY ,gB;MA4BH,Q;MAAA,OA0wET,SA1wES,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CA0wEF,qB A1wEa,OA0wEb,CA1wEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;MA0wE3D,OAzwEO,W;K;IA4wEX,sC;MAII,YA AqB,UAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAy,SAAL,SAAK,C;MA/yET,kBAAY,gB;MA4B H,Q;MAAA,OAoxET,SAPxES,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAoxEF,qBAPxEa,OAo xEb,CAPxEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAoxE3D,OAAnxEo,W;K;8FAsxEX,yB;MAAA,8C;MAAA,qC; QAKI,OAAO,iBAAM,OAAN,C;O;KALX,C;0FAQA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,Y AAY,gB;QACZ,aAAa,gB;QACG,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ, C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C; O;KAjBX,C;IAoBA,kC;MAII,IAAI,oCAAJ,C;QAAwB,OAAy,OAAL,SAAK,EAAC,OAAL,C;MACpC,aAAa,gB; MACN,OAAP,MAAO,EAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,M;K;IAGX,oC;MAII,aAA a,iBAAa,iBAAO,CAAP,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO, M;K;IAGX,qC;MAII,IAAI,oCAAJ,C;QAAwB,OAAy,OAAL,SAAK,EAAC,QAAL,C;MACpC,aAAa,gB;MACN, OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,aAA a,iBAAa,SAAK,KAAL,GAAY,QAAS,OAAR,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,E AAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,IAAI,oCAAJ,C;QAAwB,OAAy,OAAL,SAAK,EAAC,QAAL ,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OA AO,M;K;IAGX,qC;MAII,IAAI,mCAAJ,C;QACI,aAAa,iBAAa,SAAK,KAAL,GAAY,QAAS,KAAR,IAAb,C;QA Cb,MAAO,gBAAO,SAAP,C;QACP,MAAO,gBAAO,QAAP,C;QACP,OAAO,M;;QAEp,eAAa,iBAAa,SAAb,C;Q ACN,OAAP,QAAP,OAAO,QAAP,C;QACP,OAAO,Q;;K;IAIf,qC;MAII,aAAa,gB;MACN,OAAP,MAAO,EAAO, SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,aAAa,iBAAa,SAAK,KAAL ,GAAY,EAAY,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO, M;K;4FAGX,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;8FAQA,yB;MAAA,4C;MA AA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;IAQA,yD;MAGB+C,oB;QAAA,OAAy,C;MAAG,8B;QAAA, iBAA0B,K;MAOzE,Q;MANX,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eA Ae,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC, CAAnD,K;QACrB,aAAa,iBAAmB,cAAAnB,C;QACb,gBAAY,CAAZ,C;QACA,Y;UAAO,c;UAAP,MAAGB,CAAT ,mBAAiB,QAAXB,E;YAAA,K;UACI,iBAASB,eAAL,IAAK,EAAa,WAAW,OAAX,IAAb,C;UACtB,IAAI,aAAa,I AAb,IAAqB,CAAC,cAA1B,C;YAA0C,K;UhB3mGID,WAAW,iBgB4mGa,UhB5mGb,C;UaCX,mBAAC,CAAd,Y G2mGwB,UH3mGxB,Y;YbA6B,egB2mGS,sBH1mG3B,OG0mGgC,GAAK,OAAL,IAAL,ChB3mGT,C;;UgB2mG rB,MAAO,WhB1mGR,IgB0mGQ,C;UACP,oBAAS,IAAT,I;;QAEJ,OAAO,M;;MAEX,eAAa,gB;MACiE,kBAA9E, iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAY,C,cAAzC,EAAuE,KAAvE,C;ME3uGA,OAAGB,qBAAh B,C;QAAGB,gC;QF4uGL,mBE5uGqB,OF4uGrB,C;;MAEX,OAAO,Q;K;IAGX,sE;MAkBkD,oB;QAAA,OAAy,C; MAAG,8B;QAAA,iBAA0B,K;MACvF,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;Q ACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B, GAAiC,CAAnD,K;QACrB,aAAa,iBAAa,cAAAb,C;QACb,eAAa,kBAAC,SAAd,C;QACb,YAAy,C;QACZ,OAAGB, CAAT,qBAAiB,QAAXB,C;UACI,iBAASB,eAAL,IAAK,EAAa,WAAW,KAAX,IAAb,C;UACtB,IAAI,CAAC,cAA D,IAAmB,aAAa,IAApC,C;YAA0C,K;UAC1C,QAAO,cAAK,KAAL,EAAY,QAAQ,UAAAR,IAAZ,C;UACP,MAA

O,WAAI,UAAU,QAAV,CAAJ,C;UACP,gBAAS,IAAT,I;;QAEJ,OAAO,M;;MAEX,eAAa,gB;MACgE,kBAA7E,i  
BAAiB,oBAAjB,EAA6B,IAA7B,EAAMc,IAAnC,EAAYc,cAAzC,EAAuE,IAAvE,C;MErxGA,OAAgB,qBAAhB,  
C;QAAgB,gC;QFsxGL,mBAAI,UETxGiB,OFsxGjB,CAAJ,C;;MAEX,OAAO,Q;K;IAGX,kC;MAqBoB,gB;MAHh  
B,gBAXW,KAWW,O;MACtB,WAAW,iBFzkGJ,MAAO,KEykGgB,mCAAwB,EAAXB,CFzkGhB,EEykG6C,SFzk  
G7C,CEykGH,C;MACX,QAAQ,C;MACQ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT  
,C;UAAoB,K;QACpB,IAAK,WAhBqB,GAgBP,OAhBO,EAAnB,KAgBqB,CAAM,UAAN,EAAM,kBAAN,SAhB  
F,CAGrB,C;;MAhBT,OAKBO,I;K;+EAfX,yB;MAAA,kF;MAAA,gE;MFtkGA,iB;MEskGA,8C;QAWoB,UAESB,  
M;QALtC,gBAAgB,KAAM,O;QACtB,WAAW,eFzkGJ,MAAO,KEykGgB,mCAAwB,EAAXB,CFzkGhB,EEykG6  
C,SFzkG7C,CEykGH,C;QACX,QAAQ,C;QACQ,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,  
SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,OAAV,EAAMB,MAAM,UAAN,EAAM,kBAAN,SAAnB,CAA  
J,C;;QAET,OAAO,I;O;KAFx,C;IAkBA,kC;MAkBI,YAA Y,oB;MACZ,aAZW,KAYQ,W;MACnB,WAAW,iBFtm  
GJ,MAAO,KEsmGgB,mCAAwB,EAAXB,CFtmGhB,EEsmGmD,wBAbtD,KAAsD,EAAwB,EAAXB,CFtmGnD,CE  
smGH,C;MACX,OAAO,KAAM,UAAN,IAAmB,MAAO,UAAjC,C;QACI,IAAK,WafqB,GAeP,KAAM,OafC,EA  
eO,MAAO,Oafd,CAerB,C;;MAfT,OaiBO,I;K;+EAdX,yB;MAAA,kF;MAAA,gE;MFlmGA,iB;MEkmGA,8C;QA  
QI,YAA Y,oB;QACZ,aAAa,KAAM,W;QACnB,WAAW,eFtmGJ,MAAO,KEsmGgB,mCAAwB,EAAXB,CFtmGhB  
,EEsmGmD,wBAAN,KAAM,EAAwB,EAAXB,CFtmGnD,CEsmGH,C;QACX,OAAO,KAAM,UAAN,IAAmB,MA  
AO,UAAjC,C;UACI,IAAK,WAAI,UAAU,KAAM,OAAhB,EAAwB,MAAO,OAA/B,CAAJ,C;;QAET,OAAO,I;O;  
KAdX,C;IAiBA,gC;MASW,sB;;QAaP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAA d,C;UAAyB,qBAAO,W;UAAP,u  
B;;QACzB,ahB9yGoD,gB;QgB+yGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UACI,WAAW,QAAS,O  
;UACpB,MAAO,WAnBkB,GAmBJ,OAnBI,EAmBK,IANBL,CAmBIB,C;UACP,UAAU,I;;QAE d,qBAAO,M;;MA  
tBP,yB;K;8FAGJ,yB;MAAA,qD;MhBxyGA,+D;MgBwyGA,uC;QAUI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAA  
d,C;UAAyB,OAAO,W;QAC hC,ahB9yGoD,gB;QgB+yGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UA  
CI,WAAW,QAAS,O;UACpB,MAAO,WAAI,UAAU,OAAV,EAAMB,IAAnB,CAAJ,C;UACP,UAAU,I;;QAE d,OA  
AO,M;O;KAnBX,C;IAsBA,8F;MAQ6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UA  
AwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGtN,Q;MAFhB  
,MAAO,gBAAO,MAAP,C;MACP,YAA Y,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,iCA  
AU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBA  
AP,MAAO,EAAC,OAAd,EA AuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAi  
C,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4F;MAQwC,yB;QAA  
A,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QA  
AA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EA AwB,SAAXB,EAAMC,MAAnC,  
EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;4FAG5F,qB;MAKI,OAAO,S;K;I  
ASS,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAN3B,iC;MAMI,oCAAgB,8BAAhB,C;K;IAGJ,+B;MAOoB,Q  
;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QAC  
P,oBAAmB,qBAAnB,EAAMB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA O,IAAvB,GAAgC,MAA  
M,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QA  
AgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,  
wCAA O,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;  
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAnB,E;;MAEJ,  
OAAW,UAAS,CAAb,GAAgB,wCAA O,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;  
MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAn  
B,EAAMB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA O,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;M  
AOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAA  
O,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAA O,IAAvB,GAA  
gC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAh  
B,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,G  
AAgB,wCAA O,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAA  
gB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2

B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB, UA Ae,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,2 B;MAMoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,cAAO,OAAP,C;;MAEJ,OA AO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA AO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QA AgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IGI+GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,E AAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,EA AW,C AAX,CAAX,IAA4B,CAAhC,GAAMC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;MACV,wB AAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CA AjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;IA+GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,E AAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,EA AW,C AAX,CAAX,IAA4B,CAAhC,GAAMC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;MACV,wB AAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CA AjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;oGCnXX,yB;MAAA,iE;MAAA,uC;QAS8C,IAAnC,I;QAAA,+B; ;UAYS,U;UAAA,SnBgVoE,iBAAQ,W;UmBhV5F,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAbwB,SAaX,CAAU, OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;UAGR,8BAAO,I;;QAlBA,kC;QAAA,iB;UAAmC,M AAM,gCAAUb,4DAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MAAA,OAAA,SnBgVoE,QA AQ,W;MmBhV5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OA AO,M;;MAGf,OAAO,I;K;IAGX,6B;MAII,IAAI,mBAAQ,CAAZ,C;QACI,OAAO,W;MACX,eAAe,iBAAQ,W;M ACvB,IAAI,CAAC,QAAS,UAA d,C;QACI,OAAO,W;MACX,YAA Y,QAAS,O;MACrB,IAAI,CAAC,QAAS,UAA d,C;QACI,OAAO,OnBgQiD,SmBhQ1C,KnBgQ+C,IAAL,EmBhQ1C,KnBgQoD,MAAV,CmBhQjD,C;;MACX,aA Aa,iBAAsB,cAA tB,C;MACb,MAAO,WnB8PqD,SmB9PjD,KnB8PsD,IAAL,EmB9PjD,KnB8P2D,MAAV,CmB9P rD,C;;QAEwB,kBAAhB,QAAS,O;QAAPB,MAAO,WnB4PiD,SAAK,eAAL,EAAU,iBAAV,CmB5PjD,C;;MACO, QAAT,QAAS,W;MACIB,OAAO,M;K;uFAGX,yB;MAAA,+D;MAsBA,gD;MatBA,uC;QAMW,kBAAU,gB;QAo BD,Q;QAAA,OnByRoE,iBAAQ,W;QmBzR5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArB6B,SAqBIB,CAAU,O AAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QATbHb,OA wBO,W;O;KA9BX,C;uFASA,yB;MAAA,+D;MAwB A,gD;MAxBA,uC;QAUW,kBAAU,gB;QASBD,Q;QAAA,OnB0QoE,iBAAQ,W;QmB1Q5F,OAAgB,cAAhB,C;UA AgB,yB;UACZ,WAvB6B,SAuBIB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QAxBhB,OA0BO,W; O;KApCX,C;2FAaA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,OAAA,SnByRoE,QAAQ,W;QmBzR5F,OAAgB ,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QAEhB,OAAO, W;O;KARX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,OAAA,SnB0QoE,QAAQ,W;QmB1Q5F,OA AgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QAEhB,OA AO,W;O;KAZX,C;8EAeA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,cAAb,C;QA2BA,Q;QAAA,OnBiOu E,iBAAQ,W;QmBjO5F,OAAa,cAAb,C;UAAa,sB;UACT,WAA Y,WA5BiB,SA4Bb,CAAU,IAAV,CAAJ,C;;QA5B hB,OA6BO,W;O;KApCX,C;4FAUA,yB;MAAA,+D;MAAA,uC;QAOW,kBAaA,gB;QAgFJ,Q;QAAA,OnBkKoE,i BAAQ,W;QmBIK5F,OAAgB,cAAhB,C;UAAgB,yB;UAXEK,U;UAAA,cARe,SAQf,CAwEQ,OAXER,W;YAA sC,6 B;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QA6EoB,Q;QAAA,OnBkKoE,iBAAQ,W;QmBIK5F, OAAgB,cAAhB,C;UAAgB,yB;UAXEK,U;UAAA,wBAwEQ,OAXER,W;YAA sC,6B;;QAC3D,OAAO,W;O;KAN X,C;kFASA,6C;MAKiB,Q;MAAA,OAAA,SnBiOuE,QAAQ,W;MmBjO5F,OAAa,cAAb,C;QAAa,sB;QACT,WAA Y,WAAI,UAAU,IAAV,CAAJ,C;;MACHb,OAAO,W;K;8EAGX,gC;MAW oB,Q;MADhB,IAAI,mBAAJ,C;QA Ae, OAAO,I;MACN,OAAA,SnBiNoE,QAAQ,W;MmBjN5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UA AU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACTD,OAAO,I;K;IAGX,2B;MAMI,OAAO,CAAC,mB;K;+EAGZ,gC; MAO oB,Q;MADhB,IAAI,mBAAJ,C;QA Ae,OAAO,K;MACN,OAAA,SnB6LoE,QAAQ,W;MmB7L5F,OAAgB,cA AhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,qB;M AKI,OAAO,c;K;mFAGX,gC;MAMoB,Q;MAFhB,IAAI,mBAAJ,C;QA Ae,OAAO,C;MACTB,YAA Y,C;MACI,OA AA,SnB2KoE,QAAQ,W;MmB3K5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAw B,qB;;MAC9C,OAAO,K;K;sFAGX,6B;MAKoB,Q;MAAA,OAAA,SnBkKoE,QAAQ,W;MmBIK5F,OAAgB,cAAh B,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kFAG1B,yB;MJ+qDA,sE;MI/qDA,sC;QAYmB,kBAAR,iB;QAAQ,g



B;;UJ8qDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,MAAM,6B;UAC/B,cAAc,QAAS,O;UACvB,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIrrDqB,QJkrDN,CAAS,OAAT,C;;YAEX,QAAQ,QAAS,O;YACjB,QIrrDiB,QJqrDT,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAED,QAAT,QAAS,W;UACIB,eAAO,O;;QI3rDP,mB;O;KAZJ,C;8FAeA,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJ0rDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eI9rD2B,QJ8rDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIjsDuB,QJisDf,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;MIvsDP,yB;K;mFAGJ,yB;MJusDA,sE;MFn3DA,iB;MM4KA,sC;QJotDI,eIvsDO,iBJusDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIzsDqB,QJysDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI3sDiB,QJ2sDT,CAAS,QAAS,OAAIB,C;UACR,WF73DG,MAAO,KE63DO,QF73DP,EE63DiB,CF73DjB,C;;QMILd,OJ8sDO,Q;O;KI3tDX,C;mFAGBA,yB;MJ8sDA,sE;MFr5DA,iB;MMuMA,sC;QJ2tDI,eI9sDO,iBJ8sDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIhtDqB,QJgtDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIltDiB,QJktDT,CAAS,QAAS,OAAIB,C;UACR,WF/5DG,MAAO,KE+5DO,QF/5DP,EE+5DiB,C/F/5DjB,C;;QM4Md,OJqtDO,Q;O;KIluDX,C;mFAGBA,yB;MJqtDA,sE;MIrtDA,sC;QJguDI,eIrtDO,iBJqtDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIvtDqB,QJutDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIztDiB,QJytDT,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QI3tDnB,OJ8tDO,Q;O;KIzuDX,C;+FAcA,yB;MN1NA,iB;MM0NA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJ8tDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIhuD2B,QJguDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIluDuB,QJkuDf,CAAS,QAAS,OAAIB,C;YACR,WFh8DG,MAAO,KEg8DO,QFh8DP,EEg8DiB,CFh8DjB,C;;UEk8Dd,qBAAO,Q;;QIruDP,yB;O;KAXJ,C;+FAcA,yB;MnNpA,iB;MMmPA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJquDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIvuD2B,QJuuDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIzuDuB,QJyuDf,CAAS,QAAS,OAAIB,C;YACR,WFh+DG,MAAO,KEg+DO,QFh+DP,EEg+DiB,CFh+DjB,C;;UEk+Dd,qBAAO,Q;;QI5uDP,yB;O;KAXJ,C;+FAcA,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJ4uDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eI9uD2B,QJ8uDZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIhvDuB,QJgvDf,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,qBAAO,Q;;MIrvDP,yB;K;0FAGJ,yB;MJqvDA,sE;MIrvDA,kD;QJgwDI,eIrvDO,iBJqvDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIvvDqC,QJuvDtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIzvDiC,QJyvDzB,CAAS,QAAS,OAAIB,C;UACR,II1vDqB,UJ0vDN,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaK,CAAtC,C;YACI,WAAW,C;;QI3vDnB,OJ8vDO,Q;O;KIzWDX,C;sGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;QJ8vDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eIhwD2C,QJgwD5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIlwDuC,QJkwD/B,CAAS,QAAS,OAAIB,C;UACR,IIInwD2B,UJmwDZ,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaK,CAAtC,C;YACI,WAAW,C;;QAGnB,yBAAO,Q;;MIvWDP,6B;K;sFAGJ,yB;MAAA,kD;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAAR,C;O;KAVnB,C;kGAaA,yB;MAAA,8D;MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAc,UAAAd,C;O;KANnB,C;kFASA,yB;MIj4DA,sE;MIj4DA,sC;QAYmB,kBAAR,iB;QAAQ,gB;;UJg4Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,MAAM,6B;UAC/B,cAAc,QAAS,O;UACvB,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIp4DqB,QJo4DN,CAAS,OAAT,C;;YAE X,QAAQ,QAAS,O;YACjB,QIv4DiB,QJu4DT,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAED,QAAT,QAAS,W;UACIB,eAAO,O;;QI74DP,mB;O;KAZJ,C;8FAeA,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJ44Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eIh5D2B,QJg5DZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIn5DuB,QJm5Df,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;MIz5DP,yB;K;mFAGJ,yB;MJy5DA,sE;MFjhEA,iB;MMwHA,sC;QJs6DI,eIz5DO,iBJy5DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI35DqB,QJ25DN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI75DiB,QJ65DT,CAAS,QAAS,OAAIB,C;UACR,WF3hEG,MAAO,KE2hEO,QF3hEP,EE2hEiB,CF3hEjB,C;;QM6

Hd,OJg6DO,Q;O;KI76DX,C;mFAGBA,yB;MJg6DA,sE;MFnjEA,iB;MMmJA,sC;QJ66DI,eIh6DO,iBJg6DQ,W;QA Cf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI16DqB,QJk6DN,CAAS,QAAS,OAAIB,C;QACf,O AAO,QAAS,UAAhB,C;UACI,QIp6DiB,QJo6DT,CAAS,QAAS,OAAIB,C;UACR,WF7jEG,MAAO,KE6jEO,QF7j EP,EE6jEiB,CF7jEjB,C;;QMwJd,OJu6DO,Q;O;KI76DX,C;mFAGBA,yB;MJu6DA,sE;MIv6DA,sC;QJk7DI,eIv6D O,iBJu6DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIz6DqB,QJy6DN,CAAS,QAAS, OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI36DiB,QJ26DT,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAA W,CAAX,KAAJ,C;YACI,WAAW,C;;;QI76DnB,OJg7DO,Q;O;KI37DX,C;+FAcA,yB;MNtKA,iB;MMsKA,sC;QA WmB,kBAAR,iB;QAAQ,sB;;UJg7Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB; ;UACzB,eI17D2B,QJk7DZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIp7DuB,QJo7Df,CA AS,QAAS,OAAIB,C;YACR,WF9IEG,MAAO,KE8IEO,QF9IEP,EE8IEiB,CF9IEjB,C;;UEgmEd,qBAAO,Q;;;QIv7 DP,yB;O;KAXJ,C;+FAcA,yB;MN/LA,iB;MM+LA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJu7Df,eAAe,sB;UACf,I AAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIz7D2B,QJy7DZ,CAAS,QAAS,OAAIB,C;U ACf,OAAO,QAAS,UAAhB,C;YACI,QI37DuB,QJ27Df,CAAS,QAAS,OAAIB,C;YACR,WF9nEG,MAAO,KE8nE O,QF9nEP,EE8nEiB,CF9nEjB,C;;UEgoEd,qBAAO,Q;;;QI97DP,yB;O;KAXJ,C;+FAcA,+B;MASmB,kBAAR,iB; MAAQ,sB;;QJ87Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eIh8D2B ,QJg8DZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI18DuB,QJk8Df,CAAS,QAAS,OAAIB ,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,Q;;;MIv8DP,yB;K;0FAGJ,yB;MJu8 DA,sE;MIv8DA,kD;QJk9DI,eIv8DO,iBJu8DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/ B,eIz8DqC,QJy8DtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI38DiC,QJ28DzB,CAAS, QAAS,OAAIB,C;UACR,II58DqB,UJ48DN,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAA W,C;;;QI78DnB,OJg9DO,Q;O;KI39DX,C;SGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;;QJg9Df,eAAe,sB;QACf,I AAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;;QACzB,eI19D2C,QJk9D5B,CAAS,QAAS,OAAIB,C;Q ACf,OAAO,QAAS,UAAhB,C;UACI,QIp9DuC,QJo9D/B,CAAS,QAAS,OAAIB,C;UACR,Iir9D2B,UJq9DZ,SAA Q,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,yBAAO,Q;;;MIz9DP,6B;K;sFAG J,yB;MAAA,kD;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAR,C;O;KAVnB,C;kGAaA,yB;MAAA,8D; MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAC,UAAAd,C;O;KANnB,C;IASA,4B;MAMI,OAAO,mB;K;iFAGX,gC; MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAe,OAAO,I;MACN,OAAA,SnBjLoE,QAAQ,W;MmBiL5F,OAAgB,cAA hB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;MACrD,OAAO,I;K;oFAGX,6B;MA KmC,Q;MAAA,OnBILqD,iBAAQ,W;MmB0L7E,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAAr C,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MJ4wCA,wE;MI5wCA,2BAQiB,yB;QJowCjB,wE;eIpwCiB,0B;UAAA ,4B;YAAU,kBAAR,iB;YAAQ,aAAe,c;YJ2wCzB,gB;YADb,YAAy,C;YACC,6B;YAAb,OAAa,cAAb,C;cAAa,sB; cAAM,OAAO,oBAAmB,cAAnB,EAAMB,sBAAnB,UAAP,EAAoC,IAApC,C;;YI3wC2B,W;W;S;OAAjC,C;MAR jB,oC;QJmxCiB,gB;QADb,YAAy,C;QACC,OI3wCE,iBJ2wCF,W;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAA O,oBAAmB,cAAnB,EAAMB,sBAAnB,UAAP,EAAoC,IAApC,C;;QI3wCnB,gB;O;KARJ,C;4FAWA,qB;MAKI,O AAO,iB;K;IAGX,iC;MAII,OAAe,aAAR,iBAAQ,C;K;IChkBnB,kC;MAEI,gBCmE2D,8BAAY,c;MDIEvE,IAAI,S AAU,OAAV,GAAMB,CAAvB,C;QACW,Q;QAAA,IAAI,cAAQ,GAAZ,C;UAAA,OAAsB,S;;uBAAe,qBAAU,CA AV,C;UAAA,YAAe,SE0Oc,WF1OM,CE0ON,CAXCf,c;UFIMnD,OG8MoD,2BAAL,GAakB,K;;QH9MxE,W;;MA EJ,OAAuB,oBAAhB,wBAAGB,C;K;IzBD3B,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAC,SAAd,e AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAC,SAAd,e AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAC,SAAd,e AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGz C,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf, GAAyB,SAAK,M;K;IAGzC,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAC,SAAd,eAAvB,C;MACV, OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAC,SAAd,eAAvB,C;MACV,OA AO,SAAK,K;K;IAGhB,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,m BAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;gFA GzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MAAA,mC;



N,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAC,K;K;oFAG7E,yB;MAAA,6C;MAA  
A,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;oFAQA,y  
B;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;  
KALvC,C;IAQA,+B;MAOI,OAAO,sCAAE,yBAAgB,SAAhB,EAAYB,EAazB,EAakC,EAAIC,C;K;IAG1B,iC;M  
AOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAYB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAE,y  
BAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,sCAAE,yBAAqB,SAArB,EAAiC,  
EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAASB,EAAtB,EAA0B,EAA1  
B,C;K;IAG3B,iC;MAOI,OAAO,sCAAE,yBAAgB,SAAhB,EAASB,EAAtB,EAA0B,EAA1B,C;K;IAG1B,iC;MAOI  
,OAAO,uCAAgB,yBAAgB,SAAhB,EAAYB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAE,yBA  
AqB,SAArB,EAA8B,EAA9B,EAakC,EAAIC,C;K;IAG1B,iC;MAOI,OAAO,sCAAE,yBAAqB,SAArB,EAA8B,EA  
A9B,EAakC,EAAIC,C;K;IAG1B,iC;MAOI,OAAO,uCAAgB,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,  
M;K;IAG3B,iC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAASB,EAAtB,M;K;IAG3B,kC;MAOI,OAAO,uCAAg  
B,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,kC;MAOI,OAAO,uCAAgB,yBAAqB,oBAAL,S  
AAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,kC;MAOI,OAAO,sCAAE,yBAAgB,SAAhB,EAAYB,EAazB,EAakC,  
EAAIC,C;K;IAG1B,kC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAYB,oBAAH,EAAG,CAAzB,M;K;IAG3B,k  
C;MAOI,OAAO,sCAAE,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,kC;MAOI,OAAO,sCAAE,  
yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,+B;MAII,OAAO,sCAAE,yBAAgB,cAAhB,EAASB  
,eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG1B,gC;MAII,OAAO,uCAAgB,yBAAgB,cAAhB,EAASB,eAAtB,E  
AA8B,cAAD,aAA7B,C;K;IAG3B,gC;MAII,OAAO,uCAAgB,yBAAgB,cAAhB,EAASB,eAAtB,EAA6B,CAAC,cA  
AD,IAA7B,C;K;IAG3B,+B;MAII,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,sCAAE,yBAAgB,e  
AAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,GAAY,CAAhB,GAAMB,IAAnB,GAA6B,CAAC,IAAD,IAA1D,C;  
K;IAG1B,iC;MAII,oBAAoB,kBAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAgB,eAAhB,EAA  
uB,cAAvB,EAAiC,SAAK,KAAL,cAAy,CAAhB,GAAMB,IAAnB,GAA8B,IAAD,aAA1D,C;K;IAG3B,iC;MAII,o  
BAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAgB,eAAhB,EAAuB,cAAvB,EAAiC,S  
AAK,KAAL,GAAY,CAAhB,GAAMB,IAAnB,GAA6B,CAAC,IAAD,IAA1D,C;K;IAG3B,sC;MACI,OAAmB,IAA  
R,8BAAgC,GAApC,GAAiE,OAAL,SAAK,CAAjE,GAA+E,I;K;IAG1F,wC;MACI,OAAW,mEAAJ,GAAMe,OAA  
L,SAAK,SAAnE,GAAiF,I;K;IAG5F,wC;MA1OY,Q;MA2OR,OA3OkC,YAA1B,qBA2OW,aAAA,sCAAE,UAAf,E  
AA0B,sCAAE,UAAzC,CA3OX,kCAA0B,EA2OvB,SA3OuB,CA2O3B,GAAqE,OAAL,SAAK,CAArE,GAAMf,I;  
K;IAG9F,wC;MACI,OAAmB,UAAA,sCAAE,UAAf,EAA2B,sCAAE,UAA1C,CAAR,4BAAJ,GAA+E,OAAR,YA  
AL,SAAK,CAAQ,CAA/E,GAA6F,I;K;IAGxG,wC;MACI,OAAmB,UAAA,sCAAE,UAAf,EAA0B,sCAAE,UAAzC  
,CAAR,4BAAJ,GAA6E,OAAR,YAAL,SAAK,CAAQ,CAA7E,GAA2F,I;K;IAGtG,qC;MACI,OAAW,iFAAJ,GAA  
4D,SAAK,QAajE,GAA8E,I;K;IAGzF,uC;MACI,OAAmB,UAAc,WAAAd,EAAwC,UAAxC,CAAR,4BAAJ,GAAq  
E,YAAL,SAAK,CAArE,GAakF,I;K;IAG7F,uC;MACI,OAAmB,UAAc,WAAAd,EAAuC,UAAvC,CAAR,4BAAJ,G  
AAmE,YAAL,SAAK,CAAnE,GAAgF,I;K;IAG3F,sC;MACI,OAAmB,UAAe,mCAAf,EAA0C,mCAA1C,CAAR,4  
BAAJ,GAAuE,uBAAL,SAAK,CAAvE,GAAqF,I;K;IAGhG,wC;MACI,OAAmB,UAAe,mCAAf,EAAyC,mCAAzC  
,CAAR,4BAAJ,GAAqE,uBAAL,SAAK,CAArE,GAAMf,I;K;IAG9F,uC;MACI,OAAmB,MAAR,8BAAiC,KAARc  
,GAAMe,QAAL,SAAK,CAAnE,GAakF,I;K;IAG7F,yC;MACI,OAAW,uEAAJ,GAAqE,QAAL,SAAK,SAArE,GA  
AoF,I;K;IAG/F,yC;MACI,OAAmB,UAAA,uCAAgB,UAAhB,EAA4B,uCAAgB,UAA5C,CAAR,4BAAJ,GAAiF,Q  
AAR,YAAL,SAAK,CAAQ,CAAjF,GAAgG,I;K;IAG3G,yC;MACI,OAAmB,UAAA,uCAAgB,UAAhB,EAA2B,uC  
AAgB,UAA3C,CAAR,4BAAJ,GAA+E,QAAR,YAAL,SAAK,CAAQ,CAA/E,GAA8F,I;K;IAGzG,8B;MAMI,OAA  
O,wBAAy,EAAa,GAAG,CAAG,IAAzB,C;K;IAGX,gC;MAMI,OAAO,kBAAy,oBAAH,EAAG,CAAc,8BAAH,C  
AAG,EAA1B,C;K;IAGX,gC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAG,CAAG,IAAjC,C;K;IAGX,gC;MAMI,IAAI,MAAM,CAAV,C;Q  
AAoB,OAAO,iCAAU,M;MACrC,OAAO,yBAAiB,OAAR,EAAQ,GAAG,CAAG,CAAjB,C;K;IAGX,gC;MAMI,I  
AAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,wBAAS,EAAQ,GAAG,CAAG,IAAjB,C;K;IA  
GX,gC;MAMI,OAAO,kBAAy,oBAAH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;IAGX,gC;MAMI,IAAI,MAA  
M,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,aAAK,SAAL,EAAiB,EAAQ,GAAG,CAAG,IAAzB,C;K  
;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,aAAK,SAAL,EAAiB,EA



I,OAAO,sBAAQ,OAAR,KAAoB,C;K;IAWG,2C;MAAA,qB;QAAE,MAAM,8BAA0B,+CAA4C,aAA5C,MAA1B,C;O;K;IAR1C,uC;MAQI,OAAO,8BAAgB,KAAhB,EAAuB,yBAAvB,C;K;IAGX,4D;MAcqB,Q;MANjB,IAAI,QA AQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO,QAAS,UAAhB,C;Q ACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa, KAAb,C;K;IAGX,8C;MAcqB,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAY, C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C; UACI,OAAO,O;;MAEf,OAAO,I;K;8EAGX,gC;MASW,sB;;QA4FS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UA AgB,yB;UAAM,IA5FH,SA4FO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA 7FP,yB;K;uFAGJ,gC;MAMoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,I A3Nc,SA2NV,CAAU,OAAV,CAAJ,C;UACI,OAAO,O;;MA5Nf,OA+NO,I;K;IA5NX,6B;MAQI,eAAe,oB;MACf, IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,OAAO,QAAS,O;K;IFAGpB,yB;MAA A,iE;MAAA,uC;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAA J,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,sDAAvB,C;O;KARV,C;kGAWA,yB;MAAA,iE;MAAA,uC;QA W8C,IAAnC,I;QAAA,+B;;UAcS,U;UAAA,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAfwB,SAeX,CA AU,OAAV,C;YACb,IAAI,CAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QApBA,kC;QAAA,iB;UAAm C,MAAM,gCAAuB,iEAAvB,C;;QAAhD,OAAO,I;O;KAXX,C;8GAcA,gC;MAWoB,Q;MAAA,2B;MAAhB,OAA gB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,CAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I; K;IAGX,mC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,OAAO,QAAS,O;K;6F AGpB,gC;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C; UAAwB,OAAO,O;;MACrD,OAAO,I;K;IAGX,wC;MAOiB,Q;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C ;QAAa,sB;QACT,mBAAmB,KAAhB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ, OAAO,E;K;+FAGX,yB;MAAA,wE;MAAA,uC;QAOiB,Q;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UA Aa,sB;UACT,mBAAmB,KAAhB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAEJ,OAA O,E;O;KAbX,C;6FAGBA,yB;MAAA,wE;MAAA,uC;QAQiB,Q;QAFb,gBAAGB,E;QACHB,YAAY,C;QACC,2B;Q AAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAhB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAY,K; UACHB,qB;;QAEJ,OAAO,S;O;KAdX,C;IAiBA,4B;MAUI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI, MAAM,2BAAuB,oBAAvB,C;MACV,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS, O;MACpB,OAAO,I;K;+EAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAYoB,UAQT,M;QAVP,WAAe ,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI, OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,2 E;O;KApBX,C;IAuBA,4C;MAQiB,Q;MAFb,gBAAGB,E;MACHB,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;Q AAa,sB;QACT,mBAAmB,KAAhB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAY,K;QACHB,qB;;MAEJ, OAAO,S;K;IAGX,kC;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;2FAGX,gC;MASoB,Q;MADh B,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO ,O;;MAGf,OAAO,I;K;IAGX,8B;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAuB,o BAAvB,C;MACV,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,MAAM,gCAAyB,qCAAzB,C;MACV,OA AO,M;K;mFAGX,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAQoB,UAST,M;QAXP,aAAiB,I; QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,I AAI,KAJ,C;cAAW,MAAM,8BAAyB,mDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KA AL,C;UAAy,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,6E;O;KAjBX,C;IAoBA,oC;MAMI,eAAe,oB;MACf,I AAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,OAAO, I;MACX,OAAO,M;K;+FAGX,gC;MAQoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cA AhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAJ,C;YAAW,OAAO,I;UACIB,SAAS,O; UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAWW,Q;Mh BjXP,IAAI,EgBgXI,KAJK,ChBhXT,CAAJ,C;QACI,cgB+Wc,sD;QhB9Wd,MAAM,gCAAyB,OAAQ,WAAjC,C;; MgBgXN,UAAK,CAAL,C;QAAU,gB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb ,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,SAAxB,C;K;IAGX,wC;MAQ

I,OAAO,sBAaKB,SAAIB,EAAwB,IAAxB,EAA8B,SAA9B,C;K;IAcqE,iD;MAAA,qB;QAAE,yBAAU,EAAG,MAAb,EAAoB,EAAG,MAAvB,C;O;K;IAAkC,oC;MAAE,OAAA,EAAG,M;K;IAXzH,+C;MAWI,OAAO,yBAAqB,sBAaKB,qBAAiB,SAAjB,CAAIB,EAA0C,IAA1C,EAAGD,+BAAhD,CAArB,EAAYG,sBAAzG,C;K;oGAGX,yB;MAk1BA,wE;MAI1BA,oD;QA21BiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAh1BT,IAAI,UAg1BkB,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAh1BIB,EAglB+C,IAh1B/C,CAAJ,C;YAA2C,sBAglBQ,IAh1BR,C;;QAE/C,OAAO,W;O;KAbX,C;sGAgBA,yB;MAAA,8C;MAAA,0C;MAAA,8B;MASKb,qD;QAAA,qB;UAAE,c;S;O;MATpB,sC;QASW,Q;QAAP,OAAO,uCAAo,iCAAP,gC;O;KATX,C;OGAYA,4C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,YAAJ,C;UAAKB,WAAY,WAAl,OAAJ,C;;MACpD,OAAO,W;K;IAGX,2C;MAQI,OAAO,sBAaKB,SAAIB,EAAwB,KAAxB,EAA+B,SAA/B,C;K;IAYU,kC;MAAE,iB;K;IATvB,oC;MASW,Q;MAAP,OAAO,4CAAU,oBAAV,kC;K;IAGX,mD;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAY,WAAl,OAAJ,C;;MACvD,OAAO,W;K;4FAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAl,OAAJ,C;;MAC3D,OAAO,W;K;sFAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAl,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,8B;MAWW,Q;MhB1gBP,IAAI,EgBygBI,KAAK,ChBzgBT,CAAJ,C;QACI,cgBwgBc,sD;QhBvgBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MgBygBN,UAAK,CAAL,C;QAAU,sB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAMb,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAaKB,SAAIB,EAAwB,SAAxB,C;K;IAWA,2C;MAAA,8B;K;8CACH,Y;MACI,iBAA6B,iBAAZ,gBAAy,C;MACIB,QAAX,UAAW,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6B;MAQI,0C;K;sFASJ,yB;MAAA,sD;MdlfA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;Mc2ef,sC;QAUI,OAAO,sBdrfP,eAAW,iBcqfiB,QdrfjB,CAAX,CcqfO,C;O;KAVX,C;OGAAa,yB;MAAA,sD;Md5eA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;Mcqef,sC;QAQI,OAAO,sBd7eP,eAAW,2Bc6e2B,Qd7e3B,CAAX,Cc6eO,C;O;KARX,C;IAWA,uC;MAQI,OAAO,wBAAW,cAAX,C;K;IAWA,uE;MAAA,sC;MAAA,4C;K;kDACH,Y;MACI,iBAAiC,iBAAhB,oBAAGB,C;MACtB,WAAX,UAAW,EAAS,uBAAT,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6C;MAQI,0D;K;wFASJ,yB;MAAA,wE;MAAA,uC;QAaW,kBAAY,oB;QAIhF,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAlFsC,SAkFvB,CAAU,OAAV,C;UzBhEnB,wBAAl,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QyBIBa,OAoFO,W;O;KAjGX,C;6FAGBA,yB;MAAA,wE;MAAA,yC;QAaW,kBAAc,oB;QA8BL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA/B4B,WA+BxB,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA/BhB,OAiCO,W;O;KA9CX,C;6FAGBA,yB;MAAA,wE;MAAA,yD;QAYW,kBAAc,oB;QAIcL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aAIC4B,WakCxB,CAAY,OAAZ,CAAJ,EAlCyC,cAkCf,CAAE,OAAf,CAA1B,C;;QAIC hB,OAoCO,W;O;KAhDX,C;iGAeA,+C;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,YAAy,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+D;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;4FAGX,6C;MAWoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,UAAU,OAAV,C;QzBhEnB,wBAAl,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;MyBkEA,OAAO,W;K;gGAGX,yB;MAAA,wE;MAAA,2C;QAcI,aAAa,oB;QAGBG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UafO,MAGBP,aAAI,OAAJ,EAhBe,aAgBF,CAAc,OAAd,CAAb,C;;QAhBhB,OAauB,M;O;KAF3B,C;oGAKBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAMiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAl,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAMI,OAAO,0BAAa,cAAb,C;K;IAGX,8B;MAMI,OAA4B,qBAAhB,iBAA L,SAAK,CAAGB,C;K;IAGhC,qC;MAMI,OAAO,0BAAa,gBAAb,C;K;IAGX,4B;MAQI,OAAwC,oBAAjC,0BAAa,sBAAb,CAAiC,C;K;IAG5C,0C;MAYI,OAAO,uBAAmB,SAAnB,EAAYB,SAAZB,6BAAoC,qB;;OAApC,E;K;IAGX,0C;MAQI,OAAO,uBAAmB,SAAnB,EAAYB,SAAZB,6BAAoC,qB;;OAApC,E;K;IAGX,iD;MAaI,OAAO,kBA Ae,SAAf,EAAqB,SAARb,6BAAgC,qB;;OAAhC,E;K;IAGX,iD;MAaI,OAAO,kBA Ae,SAAf,EAAqB,SAARb,6BAAgC,qB;;OAAhC,E;K;sGAGX,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAY,UAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,

EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;uGAoBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;yFAoBA,yB;MAAA,gD;MAAA,oD;QAUoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KADx,C;yFAiBA,yB;MAAA,gD;MAAA,oD;QAMoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,wE;MA6BA,+D;MA7BA,yC;QAWW,kBAAU,oB;QA6BD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA9BiD,WA8BvC,CAAY,OAAZ,C;UzB9nBP,U;UADP,YyBgoBe,WzBhoBH,WyBgoBwB,GzBhoBxB,C;UACL,IAAI,aAAJ,C;YACH,ayB8nBuC,gB;YAA5B,WzB7nBX,ayB6nBgC,GzB7nBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB0nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAhCT,OAKCO,W;O;KA7CX,C;qFAcA,yB;MAAA,wE;MAkCA,+D;MAICA,yD;QAYW,kBAAU,oB;QAKCD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAnCiD,WAmCvC,CAAY,OAAZ,C;UzBlpBP,U;UADP,YyBopBe,WzBppBH,WyBopBwB,GzBppBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBkpBuC,gB;YAA5B,WzBjpBX,ayBipBgC,GzBjpBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB8oBA,iB;UACA,IAAK,WArCyD,cAqCrD,CAAe,OAAf,CAAJ,C;;QArCT,OAuCO,W;O;KAnDX,C;yFAeA,yB;MAAA,+D;MAAA,sD;QAWoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UzB9nBP,U;UADP,YyBgoBe,WzBhoBH,WyBgoBwB,GzBhoBxB,C;UACL,IAAI,aAAJ,C;YACH,ayB8nBuC,gB;YAA5B,WzB7nBX,ayB6nBgC,GzB7nBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB0nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAEt,OAAO,W;O;KAhBX,C;yFAmBA,yB;MAAA,+D;MAAA,sE;QAYoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UzBlpBP,U;UADP,YyBopBe,WzBppBH,WyBopBwB,GzBppBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBkpBuC,gB;YAA5B,WzBjpBX,ayBipBgC,GzBjpBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB8oBA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAEt,OAAO,W;O;KAjBX,C;0FAoBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAUW,sC;QAAA,8C;O;MAVX,oDAWQ,Y;QAA6C,OAAA,oBAAGB,W;O;MAXrE,iDAYQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAZ5C,gF;MAAA,yC;QAUI,2D;O;KAVJ,C;IAGBA,sC;MASI,OAAO,yBAAqB,SAArB,EAA2B,SAA3B,C;K;IAGX,4C;MASI,OAAO,gCAA4B,SAA5B,EAAC,C;K;IAGX,mD;MASI,OAAoD,gBAA7C,gCAA4B,SAA5B,EAAC,C;SAAIC,CAA6C,C;K;4GAGxD,yB;MA2NA,wE;MA3NA,oD;QAoOiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA3NsB,U;UAAA,wBA2NT,oBAAmB,cAAAnB,EAAmB,sBAAnB,UA3NS,EA2NoB,IA3NpB,W;YAA6C,6B;;;QACHf,OAAO,W;O;KAVX,C;8FAaA,yB;MAAA,wE;MAAA,oD;QAUiB,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAW,WAAI,UAAU,oBAAmB,cAAAnB,EAAmB,sBAAnB,UAAV,EAAuC,IAAvC,CAAJ,C;;QACHb,OAAO,W;O;KAZX,C;IAeA,4C;MASI,OAA6C,gBAAtC,yBAAqB,SAArB,EAA2B,SAA3B,CAAsC,C;K;8FAGjD,yB;MAAA,oD;QAGLoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAzKK,U;UAAA,wBAyKQ,OAZKR,W;YAA5C,6B;;;QAC3D,OAAO,W;O;KARX,C;iFAWA,6C;MAOiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAW,WAAI,UAAU,IAAV,CAAJ,C;;MACHb,OAAO,W;K;IAGX,gC;MAOI,OAAO,qBAAiB,SAAjB,C;K;IACgB,6B;MAAE,S;K;IAX7B,+B;MAWI,OAAy,aAAL,SAAK,EAAW,eAAX,C;K;IAGhB,2C;MAYI,OAAO,qBAAiB,SAAjB,EAAuB,QAAvB,C;K;IAGX,mC;MASiB,Q;MADb,UAAU,sB;MACG,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,GAAl,WAAI,IAAJ,C;;MACvB,OAAO,G;K;6EAGX,gC;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,C;AAL,C;UAAyB,OAAO,K;;MACTD,OAAO,I;K;IAGX,2B;MAQI,OAAO,oBAAW,U;K;6EAGtB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;IAGX,6B;MAOoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAm,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MACTB,OAAO,K;K;iFAGX,yB;MAAA,wE;MAAA,uC;QAOoB,Q;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;QAC9C,OAAO,K;O;KARX,C;8EAWA,yC;MAYoB,Q;MADhB,kBAakB,O;MACF,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;4FAGX,yB;MAAA,wE;MAAA,gD;QAcOB,UAAiD,M;QAFjE,YAAY,C;QACZ,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAAnB,EAAmB,sBAAnB,UAAV,EAAuC,WAAvC,EAAoD,OAAPD,C;;QACpC,OAAO,W;O;KAFx,C;qFAkBA,6B;MAMoB,Q;MAAA,2B;MAAhB,



OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kGAG1B,yB;MAAA,wE;MAAA,oC;QASiB,UAAgC,  
M;QAD7C,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAMb,cAAAnB,EAAmB,sBA  
AnB,UAAP,EAAoC,IAApC,C;;O;KATvB,C;IAYA,2B;MAaI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAA  
yB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ  
huCG,MAAO,KYguCE,GZhuCF,EYguCO,CZhuCP,C;;MYkuCd,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IA  
AI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QAC  
I,QAAQ,QAAS,O;QACjB,MZlwCG,MAAO,KYkwCE,GZlwCF,EYkwCO,CZlwCP,C;;MYowCd,OAAO,G;K;IAG  
X,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACn  
B,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MA  
EvB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,  
MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAA  
S,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YAC  
I,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KA1BX,C;6FA6BA,+B;MASI,eAAe  
,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,  
UAAAd,C;QAAyB,OAAO,O;MACHC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAA  
T,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,  
OAAO,O;K;iFAGX,yB;MAAA,sE;MZ/0CA,iB;MY+0CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;  
UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,  
QAAS,OAAIB,C;UACR,WZ31CG,MAAO,KY21CO,QZ31CP,EY21CiB,CZ31CjB,C;;QY61Cd,OAAO,Q;O;KatB  
X,C;iFAyBA,yB;MAAA,sE;MZn3CA,iB;MYm3CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAA  
yB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,  
OAAIB,C;UACR,WZ/3CG,MAAO,KY+3CO,QZ/3CP,EY+3CiB,CZ/3CjB,C;;QYi4Cd,OAAO,Q;O;KatBX,C;iF  
AyBA,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,  
eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAA  
I,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KatBX,C;6FAyBA,yB;MZ15CA,iB;MY05CA  
,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;  
QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZp6CG,MAAO,KY06CO,QZp6  
CP,EY06CiB,CZp6CjB,C;;QYs6Cd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ57CA,iB;MY47CA,sC;QAaI,eAAe,o  
B;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QA  
AS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZt8CG,MAAO,KYs8CO,QZt8CP,EYs8CiB,CZt8C  
jB,C;;QYw8Cd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAy  
B,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,  
OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;  
MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,  
OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QA  
AR,EAAkB,CAAI,CAAX,GAAC,CAAT,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KatBX,C;qGAyBA,2C;M  
AWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MA  
Cf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,C  
AAIB,CAAX,GAAC,CAAT,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,  
CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QA  
AQ,QAAS,O;QACjB,MZrhDG,MAAO,KYqhDE,GZrhDF,EYqhDO,CZrhDP,C;;MYuhDd,OAAO,G;K;IAGX,iC;  
MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO  
,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZnjDG,MAAO,KYmjDE,GZnjDF,EYmjDO,CZnjDP,C;;MY  
qjDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAA  
U,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UA  
Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MA  
AM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAA  
W,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MA

OI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,Q  
AAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj  
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QA  
AyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ  
l6CG,MAAO,KYk6CE,GZl6CF,EYk6CO,CZl6CP,C;;MYo6Cd,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IAA  
I,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,  
QAAQ,QAAS,O;QACjB,MZp8CG,MAAO,KY8CE,GZp8CF,EY8CO,CZp8CP,C;;MYs8Cd,OAAO,G;K;IAGX,  
2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,  
OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAE  
vB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,M  
AAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAAS,  
OAAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,  
UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KA1BX,C;6FA6BA,+B;MASI,eAAe,o  
B;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,U  
AAAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,  
C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,O  
AAO,O;K;iFAGX,yB;MAAA,sE;MZjhDA,iB;MYihDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;U  
AAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,Q  
AAS,OAAIB,C;UACR,WZ7hDG,MAAO,KY6hDO,QZ7hDP,EY6hDiB,CZ7hDjB,C;;QY+hDd,OAAO,Q;O;KAtB  
X,C;iFAyBA,yB;MAAA,sE;MZrjDA,iB;MYqjDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAy  
B,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,  
OAAIB,C;UACR,WZjkDG,MAAO,KYikDO,QZjkDP,EYikDiB,CZjkDjB,C;;QYmkDd,OAAO,Q;O;KAtBX,C;iFA  
yBA,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,e  
AAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI  
,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ5IDA,iB;MY4IDA,s  
C;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;Q  
ACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZtmDG,MAAO,KYsmDO,QZtmD  
P,EYsmDiB,CZtmDjB,C;;QYwmDd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ9nDA,iB;MY8nDA,sC;QAaI,eAAe,o  
B;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QA  
AS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZxoDG,MAAO,KYwoDO,QZxoDP,EYwoDiB,CZ  
xoDjB,C;;QY0oDd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q  
AAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QA  
AS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,  
sE;MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QA  
AS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ  
,QAAR,EAAkB,CAAI,CAAX,GAaK,CAAtC,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAtBX,C;qGAYBA,2  
C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C  
;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAA  
kB,CAAI,CAAX,GAaK,CAAtC,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,I  
AAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QAC  
I,QAAQ,QAAS,O;QACjB,MZvtDG,MAAO,KYutDE,GZvtDF,EYutDO,CZvtDP,C;;MYytDd,OAAO,G;K;IAGX,i  
C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OA  
AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZrvDG,MAAO,KYqvDE,GZrvDF,EYqvDO,CZrvDP,C;;  
MYuvDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,  
UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,  
C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAy  
B,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,  
UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD

;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAA  
O,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,C  
AAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAQI,OAAO,CAAC,oBAAW,U;K;+EAGvB,gC;M  
AQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OA  
AO,K;;MACrD,OAAO,I;K;IAUI,uC;MAAA,qB;QACP,eAAO,EAAP,C;QAAA,OACA,E;O;K;IATR,sC;MAOI,O  
AAO,kBAAI,qBAAJ,C;K;IAeW,8C;MAAA,iC;QACd,eAAO,KAAP,EAAC,OAAAd,C;QAAA,OACA,O;O;K;IAXR,  
6C;MASI,OAAO,wBAAW,4BAAX,C;K;kFAMX,yB;MAAA,4F;MAAA,uC;QAeI,eAAe,SAAK,W;QACpB,IAAI,  
CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAA  
S,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KArBX,C;gGAwBA,y  
B;MAAA,4F;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;  
UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAA  
hB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D  
,C;;QAEIB,OAAO,W;O;KAtBX,C;4GAyBA,yB;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QA  
CpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACChC,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO  
,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,Q  
AAS,OAA7D,C;;QAEIB,OAAO,W;O;KATBX,C;8FAyBA,gC;MAGBI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QA  
AS,UAAAd,C;QAAyB,OAAO,I;MACHC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,  
WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;IAoBS,2I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;  
MAAA,gD;MAAA,kD;MAAA,wB;MAAA,+B;MAAA,kC;K;;;sDAAA,Y;;;;;cACZ,gB;8BAAA,iCAAM,0BAAN,  
O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;uCACKB,0B;cACF,wD;cAAhB,gB;;;cAAA,KAAgB,yBAAhB,C;gBAAA,g  
B;;;cAAgB,oC;cACZ,yBAAc,6BAAU,sBAAV,EAAuB,OAAvB,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,  
2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;;;K;IAPgB,wF;MAAA,yD;uBAAA,+H;YAAA,S;iBAAA,Q;i  
BAAA,uB;O;K;IAjBpB,sD;MAiBI,OAAO,SAAS,iDAAT,C;K;IA4BS,yJ;MAAA,wC;MAAA,6B;MAAA,yB;MAA  
A,8C;MAAA,8D;MAAA,kD;MAAA,wB;MAAA,yB;MAAA,+B;MAAA,kC;K;;;6DAAA,Y;;;;;kBAKMc,I;cAJ/C,  
gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;iCACY,C;uCACM,0B;cACF,+D;cAAhB,gB;;;  
cAAA,KAAgB,yBAAhB,C;gBAAA,gB;;;cAAgB,oC;cACZ,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAA  
nB,QAAV,EAAuC,sBAAvC,EAAoD,OAApD,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;c  
AAA,Q;;;cAFJ,gB;;;cAIJ,W;;;;;K;IARgB,sG;MAAA,yD;uBAAA,6I;YAAA,S;iBAAA,Q;iBAAA,uB;O;K;I  
AlBpB,6D;MAkBI,OAAO,SAAS,wDAAT,C;K;IA2BS,4H;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,oD;MAAA,  
kD;MAAA,4B;MAAA,+B;MAAA,kC;K;;;wDAAA,Y;;;;;oCACG,wC;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mB  
AAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;;cAGI,gB;;c  
AAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;cACI,yBAAc,6BAAU,sBAAV,EAAuB,mBAAS,OAAhC,C;cACd,  
gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAHJ,gB;;;cAQJ,W;;;;;K;IAV  
gB,yE;MAAA,yD;uBAAA,gH;YAAA,S;iBAAA,Q;iBAAA,uB;O;K;IAhBpB,+C;MAGBI,OAAO,SAAS,0CAAT,C  
;K;IA6BS,0I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,kE;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,yB;MAAA  
,kC;K;;;+DAAA,Y;;;;;cAOuC,Q;oCANpC,+C;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gC  
AAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;;iCAGgB,C;cACZ,gB;;;cAAA,KAAO  
,mBAAS,UAAhB,C;gBAAA,gB;;;cACI,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAAnB,QAAV,EAAuC,  
sBAAvC,EAAoD,mBAAS,OAA7D,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cA  
FJ,gB;;;cAJJ,gB;;;cASJ,W;;;;;K;IAXgB,uF;MAAA,yD;uBAAA,8H;YAAA,S;iBAAA,Q;iBAAA,uB;O;K;IA  
hBpB,sD;MAGBI,OAAO,SAAS,iDAAT,C;K;IAcX,+C;MAkBI,OAAO,yBAAY,OAAZ,EAAqB,SAArB,C;K;IAGX  
,sD;MAmBI,OAAO,gCAAmB,OAAnB,EAA4B,SAA5B,C;K;gFAGX,+B;MASoB,Q;MADhB,UAAe,C;MACC,2B  
;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;4FAGX,+B;M  
ASoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,SAAS,OAAT,C;;  
MAEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,y  
B;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAA  
hB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;iFAGX,yB;MAAA,S  
AWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,cAAO,S

AAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KafX,C;iFakBA,yB;M5B/jEA,6B;M4B+jEA,sC;QAaoB,Q;QADhB,U5  
BjkEmC,c4BikEnB,C5BjkEmB,C;Q4BkkEnB,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,M5B/4EiD,c4B+  
4EjD,G5B/4E2D,KAAK,G4B+4EzD,SAAS,OAAT,C5B/4EoE,KAAX,IAAf,C;;Q4B5ErD,OAAO,G;O;KAhBX,C;  
iFamBA,yB;MX/kEA,+B;MW+kEA,sC;QAaoB,Q;QADhB,UXhIEqC,eAAW,oBWglE/B,CXhIE+B,CAAX,C;QW  
iErB,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,MX/5EmD,eW+5EnD,GX/5E8D,KAAK,KW+5E5D,SA  
AS,OAAT,CX/5EuE,KAAX,CAAhB,C;;QW6EvD,OAAO,G;O;KAhBX,C;IAyBe,oD;MAAA,qB;QAAE,e;UAAM  
,MAAM,gCAAyB,2BAAwB,mBAAxB,MAAzB,C;;QAAZ,S;O;K;IANjB,qC;MAMI,OAAO,kBAAI,gCAAJ,C;K;I  
AGX,oC;MAaI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,+C;MAkBI,OAAO,sBAAS,IAAT,E  
AAe,IAAf,EAAc,IAAtC,EAAwD,SAAXD,C;K;IASA,0D;MAAA,4B;MAAA,sC;K;IAG0B,+E;MAAA,qB;QAAE  
,IAAI,CAAC,iBAAD,IAAY,WAAM,eAAN,CAAhB,C;UAAiC,oBAAU,I;UAA3C,OAAiD,K;;UAAjD,OAA8D,I;  
O;K;6CAF7F,Y;MACI,kBAAc,KAAd,C;MACA,OAakB,SAAX,eAAW,EAAO,kEAAp,CAA8E,W;K;;IAT5G,qC;  
MAMI,kD;K;IAkBO,6D;MAAA,4B;MAAA,wC;K;IAE6B,iE;MAAA,qB;QAAE,gBAAM,gBAAN,K;O;K;+CADl  
C,Y;MACI,OAakB,YAAX,eAAW,EAAU,4DAAV,CAA6B,W;K;;IAZ3D,sC;MASI,IAAI,Q/B0qKG,YAAQ,C+B1  
qKf,C;QAAwB,OAAO,S;MAC/B,qD;K;IAGBO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAA  
M,aAAN,mB;O;K;+CALtC,Y;MACI,YAAqB,6BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,e  
AAW,W;;QAEIB,OAakB,YAAX,eAAW,EAAU,4CAAV,CAA0B,W;K;;IAf5D,sC;MASI,qD;K;IAoBO,6D;MAA  
A,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAAM,aAAN,mB;O;K;+CALtC,Y;MACI,YAAqB,UAAAT,qB  
AAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,eAAW,W;;QAEIB,OAakB,YAAX,eAAW,EAAU,4CAAV,  
CAA0B,W;K;;IAf5D,sC;MASI,qD;K;8FAWJ,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,iBAAM,OAAN,C;O;KAP  
X,C;wFAUA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAYoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACG,2B;Q  
AAhB,OAAGb,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAl,OAAJ,C;;YAE  
N,MAAO,WAAl,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAy,MAAZ,C;O;KAnBX,C;IAsBA,oC;MAMI,OAA6  
C,UAAtC,YAAW,SAAX,EAAiB,YAAW,OAAX,EAAjB,EAAc,C;K;IAGjD,qC;MASI,OAAy,OAAL,SAAK,EA  
Ac,OAAT,QAAS,CAAd,C;K;IAGhB,qC;MASI,OAA+C,UAAx,C,YAAW,SAAX,EAA0B,aAAT,QAAS,CAA1B,E  
AAwC,C;K;IAGnD,sC;MASI,OAakC,UAA3B,YAAW,SAAX,EAAiB,QAAjB,EAA2B,C;K;4FAGtC,yB;MAAA,  
0C;MAAA,qC;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;IAUA,2D;MAGb+C,oB;QAAA,OAAy,C;MAAG,8B;  
QAAA,iBAA0B,K;MACpF,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA2D,KAA3D,C;K;IAGX,  
sE;MAkBkD,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,OAAwE,OAAjE,8BAAiB,IAAjB,EAA  
uB,IAAvB,EAA6B,cAA7B,EAA2D,IAA3D,CAAiE,EAAI,SAAJ,C;K;IAYpC,4B;MAAY,cAAM,EAAAN,C;K;IATp  
D,kC;MASI,OAAO,oBAAgB,SAAhB,EAAcB,KAAtB,EAA6B,UAA7B,C;K;IAGX,6C;MAUI,OAAO,oBAAgB,S  
AAhB,EAAcB,KAAtB,EAA6B,SAATB,C;K;IAcY,kC;MAAU,aAAK,CAAL,C;K;IAXjC,kC;MAWI,OAAO,yBAA  
Y,kBAAZ,C;K;IAeiB,wH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,gD;MAAA,kD;MAAA,4B;MAAA,2B;MAA  
A,wB;MAAA,kC;K;;sDAAA,Y;;oCACl,sC;cACf,IAAI,CAAC,mBAAS,UAAAd,C;gBAAyB,M;;gBAAzB,gB;;;  
;;mCACc,mBAAS,O;cACvB,gB;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;gCACe,mBAAS,O;cACpB,gB  
;8BAAA,iCAAM,6BAAU,kBAAV,EAAmB,eAAnB,CAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,qBAAU,e  
;cAHd,gB;;cAKJ,W;;K;IATwB,uE;MAAA,yD;uBAAA,4G;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAZ5B,  
6C;MAYI,OAAO,SAAS,0CAAT,C;K;IAYX,8F;MAU6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MA  
AI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;  
MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACI,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,y  
B;QACZ,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KA  
A1B,C;UACW,gBAAP,MAAO,EAAc,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QA  
AQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4  
F;MAUwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,Q  
AAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EAAwB,SA  
xB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAATB,EAAcE,SAATe,CAAiF,W;K;IAOXE,8C;  
MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAJ3B,kC;MAII,oCAAGb,8BAAhB,C;K;2FAGJ,qB;MAKI,OAAO,S;K;  
IAGX,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;  
QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAAnB,E;;MAEJ,OAAW,UAAAS,CAAb,GAAGb,wCAAO,I

AAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,O  
AAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAAnB,E;;MAEJ,OAAW,UA  
AS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YA  
AiB,C;MACD,2B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,  
KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;M  
AFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,  
oBAAmB,qBAAnB,EAAMB,KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,  
K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAGB,cAAhB,C;QAAGB  
,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCA  
AO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAh  
B,OAGB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAMB,KAAAnB,E;;MAEJ,OAAW  
,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2  
B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,U  
AAe,C;MACC,2B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQ  
oB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAA  
O,G;K;IAGX,2B;MAQoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAGB,cAAhB,C;QAAGB,yB;QACZ,cAAO,OAA  
P,C;;MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAGB,cAAhB,C;QAAGB  
,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAGB,  
cAAhB,C;QAAGB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IC/FX,qC;MAMI,aAAa,qBAAiB,YAAY,cAAZ,CA  
AjB,C;MACb,kBAAC,KAAd,C;MX8zBgB,Q;MAAA,OW7zBT,SX6zBS,W;MAAhB,OAGB,cAAhB,C;QAAGB,2  
B;QAAU,oB;QW7zBK,IAAI,CAAC,SAAD,IAAY,OX6zBX,SW7zBW,UAAhB,C;UAAiC,YAAU,I;UAA3C,mBA  
AiD,K;;UAAjD,mBAA8D,I;;QX6zBvE,qB;UW7zBD,MX6zBqC,WAAI,SAAJ,C;;MW7zB1D,OAAqB,M;K;IAGz  
B,sC;MAMI,aAAa,qBAAiB,SAAjB,C;MACN,YAAP,MAAO,EAAU,QAAY,C;MACP,OAAO,M;K;IAGX,sC;MA  
MI,YAAqB,6BAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAAY,QAAL,SAAK,C;MACb,IAAI,yBA  
AJ,C;QACgB,kBAAY,sB;QX2xBZ,Q;QAAA,OW3xBL,SX2xBK,W;QAAhB,OAGB,cAAhB,C;UAAgB,yB;UA  
AM,IAAI,CW3xBwB,qBX2xBb,OW3xBa,CX2xB5B,C;YAAyB,WAAy,WAAI,OAAJ,C;;QW3xBvD,OX4xBG,W  
;;MW3xBP,aAAa,qBAAiB,SAAjB,C;MACb,MAAO,mBAAU,KAAY,C;MACP,OAAO,M;K;IAGX,uC;MAMI,aA  
Aa,qBAAiB,SAAjB,C;MACN,YAAP,MAAO,EAAU,QAAY,C;MACP,OAAO,M;K;gGAGX,yB;MAAA,8C;MAA  
A,qC;QAOI,OAAO,iBAAM,OAAN,C;O;KAPX,C;IAUA,qC;MAMI,aAAa,qBAAiB,YAAY,iBAAO,CAAP,IAAZ,  
CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,M;K;IAGX,sC;MAOI,  
aAAa,qBAAiB,YAAY,SAAK,KAAL,GAAY,QAAS,OAArB,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;M  
ACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;MAMuD,UAAAT,M;MAA1C,aAAa,qBAAiB,  
YAAY,WAAS,4BAAT,QAAS,CAAT,YAA4C,cAAL,WAAvC,4BAA2D,SAAK,KAAL,GAAY,CAAZ,IAAvE,CA  
AjB,C;MACb,MAAO,gBAAO,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;M  
AOI,aAAa,qBAAiB,YAAY,SAAK,KAAL,GAAY,CAAZ,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MAC  
A,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;8FAGX,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,gBAAK,  
OAAL,C;O;KAPX,C;InBvHA,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3C,gF;ICGA,oD;MAQuF,  
wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;gGmBYA,yB;MAAA,uD;MAAA,gC;MAAA,iD;QAOI,OAAW,  
SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KAJ,CAATc,GAAsD,uBAAa,KAAb,E;O;KAPjE,C;gGAU  
A,yB;MAAA,+C;MAAA,mC;QAOI,OAAAY,UAAAL,SAAK,EAAU,KAAY,C;O;KAPhB,C;0EAUA,yB;MA6EA,6C;  
MAAA,oC;MAAA,gC;MA7EA,uC;QAOW,sB;;UA0ES,Q;UAAA,0B;UAAhB,OAGB,cAAhB,C;YAAgB,oC;YA  
AM,IA1EH,SA0EO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;UAC9C,qBAAO,I;;QA3EP,yB;O;K  
APJ,C;kFAUA,yB;MAyJA,mD;MAAA,+C;MAAA,oC;MAzJA,uC;QAOW,qB;;UAwJO,Q;UAAA,OAAa,SAAR,s  
BAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,qBAAK,KAAL,C;YACd,IA1Jc,SA0JV,CAAU,o  
BAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;UAE5B,oBAAO,I;;QA5JP,wB;O;KAPJ,C;IAUA,6B;MAMI,ICiO  
gD,qBAAU,CDjO1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAK,CAAL,C;K;4EAGX,yB;MA  
AA,6C;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,0B;QAAhB,OAGB,cAAhB,C;UAAgB,o  
C;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAUb,6DAAvB,C;O;KANV,C;6

FASA,yB;MAAA,iE;MAYA,6C;MAAA,oC;MAAA,gC;MAZA,uC;QAS8C,IAAnC,I;QAAA,+B;;UAYS,U;UAAA,4B;UAAhB,OAAgB,gBAAhB,C;YAAgB,sC;YACZ,aAbwB,SAaX,CAAU,oBAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIBA,kC;QAAA,iB;UAAmC,MAAM,gCAAUb,sEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;yGAYA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QASoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,aAAa,UAAU,oBAAV,C;UACb,IAAI,cAAJ,C;YACI,OAAO,M;;;QAGf,OAAO,I;O;KAFx,C;IAKBA,mC;MAII,OCiLgD,qBAAU,CDjLnD,GAAe,IAAf,GAAYb,qBAAK,CAAL,C;K;wFAGpC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAm,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;mFAQA,yB;MAAA,uD;MAAA,gC;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KAAJ,CAAtC,GAAsD,uBAAa,KAAb,E;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAsC,qBAAI,KAAJ,CAAtC,GAAsD,I;K;OFAGjE,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIkB,gC;QAAA,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;wFAYA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,4B;MAQI,ICqHgD,qBAAU,CDrH1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAK,2BAAL,C;K;0EAGX,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAUb,6DAAvB,C;O;KAZV,C;IAeA,kC;MAMI,OC2FgD,qBAAU,CD3FnD,GAAe,IAAf,GAAYb,qBAAK,mBAAS,CAAT,IAAL,C;K;sFAGpC,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;8EAaA,yB;MAAA,mC;MAAA,yC;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,ICyDgD,qBAAU,CDzD1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;0FAGX,yB;MAAA,mC;MAAA,qD;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,ICoCgD,qBAAU,CDpC1D,C;QACI,OAAO,I;MACX,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,yBAAvB,C;aACX,C;UAAK,4BAAK,CAAL,C;UAAAL,K;;UACQ,MAAM,gCAAYb,0CAAzB,C;;MAHIB,W;K;8EAOJ,yB;MAAA,6C;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,wDAAzB,C;YACjB,SAAS,O;YACT,QAQAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAUb,6DAAvB,C;QAEIB,OAAO,4E;O;KAFx,C;IAkB,A,oC;MAII,OAAW,qBAAU,CAAd,GAAiB,qBAAK,CAAL,CAAjB,GAA8B,I;K;0FAGzC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YACIB,SAAS,O;YACT,QAQAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,+B;MIB1RI,IAAI,EkBkSI,KAAK,CIBIST,CAAJ,C;QACI,ckBiSc,wD;QIBhSd,MAAM,gCAAYb,OAAQ,WAAjC,C;;MkBiSV,OAAO,8BAAc,eAAF,CAAE,EAAa,gBAAb,CAAd,EAAoC,gBAApC,C;K;IAGX,+B;MIBtSI,IAAI,EkB8SI,KAAK,CIB9ST,CAAJ,C;QACI,ckB6Sc,wD;QIB5Sd,MAAM,gCAAYb,OAAQ,WAAjC,C;;MkB6SV,OLxF6E,oBKwF1D,eAAF,CAAE,EAAa,gBAAb,CLxF0D,C;K;IK2FjF,kC;MIBITI,IAAI,EkB0TI,KAAK,CIB1TT,CAAJ,C;QACI,ckByTc,wD;QIBxTd,MAAM,gCAAYb,OAAQ,WAAjC,C;;MkByTV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,CAAd,CAAIb,C;K;IAGX,mC;MIB9TI,IAAI,EkBsUI,KAAK,CIBtUT,CAAJ,C;QACI,ckBqUc,wD;QIBpUd,MAAM,gCAAYb,OAAQ,WAAjC,C;;MkBqUV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,CAAd,CAAIb,C;K;2FAGX,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAy,CAAZ,EAAe,QAQAQ,CAAR,IAAf,C;QACf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OLpIoF,oBKoInE,CLpImE,EkolhE,QAQAQ,CAAR,ILpIgE,C;;QKqI5F,OAAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,IAAI,CAA

C,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QACf,OAAO,E;O; KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK ,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV ,CAAL,C;YACI,OL/JqE,oBK+JpD,KL/JoD,C;;QKqK7E,OAAO,E;O;KATX,C;8EAYA,yB;MAAA,yD;MAkFA,o C;MAIFA,uC;QAMW,kBAAS,oB;QakFM,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAL,KAAJ,C; UACd,IAPf6B,SAoFzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QApFxC,OAsFO,W;O;KA 5FX,C;8EASA,yB;MAAA,yD;MAyEA,oC;MAzEA,uC;QAMW,kBAAS,oB;QAYEM,Q;QAAA,uB;QAAtB,iBAAc ,CAAd,wB;UACI,cAAc,qBAAL,KAAJ,C;UACd,IA3E6B,SA2EzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBA AO,OAAP,C;;QA3ExC,OA6EO,WA7EqC,W;O;KANhD,C;4FASA,yB;MAAA,yD;MAsBA,gC;MAmtBA,6C;MA AA,oC;MAZuBA,uC;QAQW,kBAAGB,oB;QAwuBV,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAA a,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UAjtB/B,IAvBoC,SAuBhC,CAAU,OA AV,EAAiB, OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAvB/C,OAYBO,W;O;KAjCX,C;4FAWA,yB;MAAA,yD;MAWA,g C;MAmtBA,6C;MAAA,oC;MA9tBA,uC;QAQW,kBAAGB,oB;QA6tBV,gB;QADb,YAAy,C;QACC,0B;QAAb,O AAa,cAAb,C;UAAA,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UAjtB/B,IAZoC,SAyhC,CAAU ,OA AV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAZ/C,OAcO,WAd4C,W;O;KARvD,C;gGAWA,y B;MAAA,gC;MAmtBA,6C;MAAA,oC;MAntBA,oD;QA0tBiB,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAA b,C;UAAA,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UAjtB/B,IAAI,UAAU,OA AV,EAAiB,OA AjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAE/C,OAAO,W;O;KAXX,C;oFAcA,yB;MAAA,yD;MAkBA,6C;MA AA,oC;MAAA,gC;MAIBA,uC;QAMW,kBAAY,oB;QakBH,Q;QAAA,0B;QAAhB,OAGB,cAAhB,C;UAAgB,oC ;UAAM,IAAI,CAIBU,SAkBT,CAAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAlB3D,OAmBO,W ;O;KAZBX,C;oFASA,yB;MAAA,yD;MASA,6C;MAAA,oC;MAAA,gC;MATA,uC;QAMW,kBAAY,oB;QASH,Q; QAAA,0B;QAAhB,OAGB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CATU,SAST,CAAU,oBAAV,CAAL,C;YAAyB, WAAy,gBAAO,OAAP,C;;QAT3D,OAuO,WAVwC,W;O;KANnD,C;wFASA,yB;MAAA,6C;MAAA,oC;MAAA, gC;MAAA,oD;QAMoB,Q;QAAA,0B;QAAhB,OAGB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,UAAU,oBAA V,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAC3D,OAAO,W;O;KAPX,C;kFAUA,yB;MAAA,oC;MAAA,oD ;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAL,KAAJ,C;UACd,IAAI,UAAU,oBAAV,CA AJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QAExC,OAAO,W;O;KAVX,C;IAaA,sC;MAIL,IAAI,OAAQ,UAAZ,C;Q AAuB,OAAO,E;MAC9B,OAAO,yBAAY,OAAZ,C;K;IAGX,sC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,E; MAC9B,OAAO,uBAAU,OA AV,C;K;IAGX,sC;MAOc,Q;MAHV,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C; MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,E;MAcTb,aAAa,mBAAc,IAAd,C;MACH,yB;MAAV,OAAU,cAAV ,C;QAAU,mB;QACN,MAAO,gBAAO,qBAAL,CAAJ,CAAP,C;;MAEX,OAAO,M;K;4EAGX,yB;MAAA,8B;MAA A,uC;MAAA,qC;QAKY,Q;QAAR,OAA8B,MAAtB,2DAAsB,EAAM,OAAN,CAAE,W;O;KALjD,C;IAQA,+B;MI B9fI,IAAI,EkBsgBI,KAAK,CIBtgBT,CAAJ,C;QACI,ckBqgBc,wD;QIBpgBd,MAAM,gCAAYB,OAAQ,WAAjC,C; ;MkBgqBV,OAAO,8BAAY,CAAZ,EAAiB,eAAF,CAAE,EAAa,gBAAb,CAAjB,C;K;IAGX,+B;MIB1gBI,IAAI,Ek BkhBI,KAAK,CIBlhBT,CAAJ,C;QACI,ckBihBc,wD;QIBhhBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBiBV,O LzT4F,oBKyt3E,CLzT2E,EKyTtE,eAAF,CAAE,EAAa,gBAAb,CLzTsE,C;K;IK4ThG,kC;MIBthBI,IAAI,EkB8hB I,KAAK,CIB9hBT,CAAJ,C;QACI,ckB6hBc,wD;QIB5hBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkB6hBV,aAAa, gB;MACb,OAAO,8BAAY,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,IAAZ,EAA6C,MAA7C,C;K;IAGX,mC;MI BniBI,IAAI,EkB2iBI,KAAK,CIB3iBT,CAAJ,C;QACI,ckB0iBc,wD;QIBziBd,MAAM,gCAAYB,OAAQ,WAAjC,C; ;MkB0iBV,aAAa,gB;MACb,OLtV6E,oBKsv5D,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,ILtV4D,C;K;2FKyVj F,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iC AAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,QAAQ,CAAR,IAAZ,EAAuB,gBAAvB,C;;QAGf,OAAO,8B AAY,CAAZ,EAAe,gBAAf,C;O;KAXX,C;4FAcA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd, WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OL/WqE,oBK+WpD,QAA Q,CAAR,IL/WoD,C;;QKkX7E,OAAO,S;O;KAXX,C;oFAcA,yB;MAAA,oC;MAAA,uC;QAM0B,Q;QAAA,uB;Q AAAtB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAL,KAAJ,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ, EAAe,KAAf,C;;QAEf,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;O;KAVX,C;oFAaA,yB;MAAA,oC;MAAA,uC;QA M0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAI,KAAJ,EAAV,CAAL,C;YACI,O

LvYoF,oBKuYnE,CLvYmE,EKuYhE,KLvYgE,C;;QKyY5F,OAAO,S;O;KAVX,C;IAaA,gC;MAII,OAAO,qBAAc,SAAd,CAAoB,U;K;kFAG/B,yB;MAAA,8B;MAAA,6C;MAAA,4B;QAKY,Q;QAAR,OAA8B,SAAtB,2DAAsB,C AAW,W;O;KAL7C,C;oFAQA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA4EA,6C;MAAA,oC;MAAA,gC;MA5EA ,uC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAC,EAAd,C;QAC5B,kBAAY,mBAAoB,QAAPB,C;QAYE H,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WA1E8C,SA0E/B,CAAU,oBAAV,C;U3B3EnB,w BAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;Q2BCA,OA4EO,W;O;KAXFX,C;wFAeA,yB;MAAA,0D;MAAA,y D;MAAA,uE;MA6BA,6C;MAAA,oC;MAAA,gC;MA7BA,yC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,E AAc,EAAd,C;QAC5B,kBAAc,mBAAuB,QAavB,C;QA2BL,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC ;UACZ,WAAy,aA5BuC,WA4BnC,CAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C ;wFAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA8BA,6C;MAAA,oC;MAAA,gC;MA9BA,yD;QAUI,eAAmC,c AApB,YAAY,gBAAZ,CAAoB,EAAC,EAAd,C;QAC5B,kBAAc,mBAAoB,QAAPB,C;QA6BL,Q;QAAA,0B;QAA hB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAy,aA9BoC,WA8BhC,CAAY,oBAAZ,CAAJ,EA9BiD,cA8BvB,C AAe,oBAaf,CAA1B,C;;QA9BhB,OAgCO,W;O;KA3CX,C;4FAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA, sD;QAuOB,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAy,aAAI,YAAY,oBAAZ,CAAJ,EA A0B,oBAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;4FAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sE;QAU oB,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAy,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,eA Ae,oBAaf,CAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;wFAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD; QASoB,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAe,UAAU,oBAAV,C;U3B3EnB,wBAAI ,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;Q2B6EA,OAAO,W;O;KAZX,C;4FAeA,yB;MAAA,uD;MAAA,0D;MA AAA,yD;MAAA,uE;MAGBA,6C;MAAA,oC;MAAA,gC;MAhBA,2C;QAYI,aAAa,mBAA6D,cAAAtC,YAAmB,aAA P,gBAAO,EAAa,GAAb,CAAnB,CAAsC,EAAC,EAAd,CAA7D,C;QAcG,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C ;UAAgB,oC;UAbO,MAcP,aAAI,oBAAJ,EAd,e,aAcF,CAAc,oBAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;+F AgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,wD;QAuOB,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAA gB,oC;UACZ,WAAy,aAAI,oBAAJ,EAAa,cAAc,oBAAd,CAAb,C;;QAEhB,OAAO,W;O;KAbX,C;IAGBA,iD;MA IiB,Q;MAAA,4B;MAAb,OAAa,cAAb,C;QAAa,iC;QACT,WAAy,WAAI,iBAAJ,C;;MAEhB,OAAO,W;K;IAGX,i C;MAII,OAAO,2BAaA,eAAc,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB,CAAd,CAAb,C;K;IAGX,8B;MAIiB ,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,iCAAK,CAAL,EAAP,C ;UAAL,K;;UACa,wBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,qC;MAII,OAAO,2BAaA,iBAAgB,gBAAhB,C AAb,C;K;IAGX,6B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,C;UAAK,aA AM,iCAAK,CAAL,EAAN,C;UAAL,K;;UACQ,kCAAa,qBAAoB,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB, CAAPB,CAAb,C;UAHL,K;;MAAP,W;K;gFAOJ,yB;MAAA,+D;MA0CA,6C;MAAA,oC;MAAA,gD;MAAA,gC; MA1CA,uC;QAMW,kBAAU,gB;QAwCD,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAZC6B, SAyCIB,CAAU,oBAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA1ChB,OA4CO,W;O;KAIDX,C;8FASA,yB; MAAA,+D;MAeA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAfA,uC;QAYW,kBAAiB,gB;QAcR,gB;QADhB,YAA Y,C;QACI,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAFoC,SAezB,EAAU,cAAV,EAAU,sBAAV,WAA mB,oBAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAhBhB,OAkBO,W;O;KA9BX,C;kGAeA,yB;MAAA,6C ;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,0B;QAaHb,OAAgB,c AAhB,C;UAAgB,oC;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAAnB,C;UACC,OAAZ,WAAy, EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;oFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAA A,oD;QAIoB,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,UAAU,oBAAV,C;UACC,OA AZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;gFAWA,yB;MAAA,wE;MAyBA,6C;MAAA,oC;MA AA,+D;MAAA,gC;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,0B;QAaHb,OAAgB,cAAhB,C;UAAgB,oC ;UACZ,UA1BoD,WA0B1C,CAAY,oBAAZ,C;U3BljBP,U;UADP,Y2BojBe,W3BpjBH,W2BojBwB,G3BpjBxB,C; UACL,IAAI,aAAJ,C;YACH,a2BkjBuC,gB;YAA5B,W3BjjBX,a2BjjBgC,G3BjjBhC,EAAS,MAAT,C;YACA,e;;Y AEA,c;;U2B8iBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5BT,OA8BO,W;O;KAvcX,C;gFAYA,yB;MAAA,wE;M A8BA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,0B;QAaHb,OA AgB,cAAhB,C;UAAgB,oC;UACZ,UA/BiD,WA+BvC,CAAY,oBAAZ,C;U3BpkBP,U;UADP,Y2BskBe,W3BtkBH ,W2BskBwB,G3BtkBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BokBuC,gB;YAA5B,W3BnkBX,a2BmkBgC,G3BnkBh



C,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2BgkBA,iB;UACA,IAAK,WajCyD,cAiCrD,CAAe,oBAAf,CAAJ,C;;QA  
jCT,OAmCO,W;O;KA7CX,C;oFAaA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QA  
AA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;U3BljBP,U;UADP,Y2BojBe,W3  
BpjBH,W2BojBwB,G3BpjBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BkjBuC,gB;YAA5B,W3BjjBX,a2BijBgC,G3Bjj  
BhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2B8iBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAd  
X,C;oFAiBA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAAhB,OAAg  
B,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;U3BpkBP,U;UADP,Y2BskBe,W3BtkBH,W2BskBwB,  
G3BtkBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BokBuC,gB;YAA5B,W3BnkBX,a2BmkBgC,G3BnkBhC,EAAS,MA  
AT,C;YACA,e;;YAEA,c;;U2BgkBA,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C  
;qFakBA,yB;MAAA,6C;MAAA,oC;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,  
Y;QAAgD,OAAgB,SAAhB,oBAAgB,C;O;MATxE,iDAUQ,mB;QAAuC,gCAAY,oBAAZ,C;O;MAV/C,gF;MAA  
A,yC;QAQI,2D;O;KARJ,C;wEAcA,yB;MAAA,gE;MAyEA,6C;MAAA,oC;MAAA,gC;MAZEa,uC;QAOW,kBAA  
M,eAAa,gBAAb,C;QAUeA,Q;QAAA,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAXEmB,SAwEf,CAA  
U,iBAAV,CAAJ,C;;QAxehB,OAYEO,W;O;KAhFX,C;sFAUA,yB;MAAA,gE;MA+BA,6C;MAAA,oC;MAAA,gC;  
MA/BA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAA  
a,iC;UACT,WAAY,WajC0B,SAiCtB,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QajChB,OAkC  
O,W;O;KAZCX,C;mGAUA,yB;MAAA,+D;MAUA,gC;MAwLA,6C;MAAA,oC;MAIMA,uC;QAOW,kBAAoB,gB;  
QAKMd,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAXLSB,U;UAAA,cAVQ,SAUR,EAwL  
T,cAxLS,EAwLT,sBAxLS,WAwLA,iBAxLA,W;YAA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;uGAUA,yB;MAA  
A,gC;MAwLA,6C;MAAA,oC;MAxLA,oD;QA+LiB,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,  
iC;UAXLSB,U;UAAA,yBAwLT,cAxLS,EAwLT,sBAxLS,WAwLA,iBAxLA,W;YAA6C,6B;;;QACHF,OAAO,W;O  
;KARX,C;oFAWA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAQiB,UACiB,M;QAF9B,YAAY,C;QACC  
,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CA  
AJ,C;;QACHB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,+D;MAUA,gC;MA+IA,6C;MAAA,oC;MAZJA,uC;QAO  
W,kBAAa,gB;QAsJJ,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA9IK,U;UAAA,cARe,SAQf,CA8IQ,  
oBA9IR,W;YAA5C,6B;;;QAR3D,OASO,W;O;KAhBX,C;yFAUA,yB;MAAA,gC;MA+IA,6C;MAAA,oC;MA/IA,o  
D;QAmJoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA9IK,U;UAAA,wBA8IQ,oBA9IR,W;YAA5C,  
6B;;;QAC3D,OAAO,W;O;KANX,C;4EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAKiB,Q;QAAA,  
0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAAI,UAAU,iBAAV,CAAJ,C;;QACHB,OAAO,W;O;KAPX,  
C;IAe4B,4C;MAAA,mB;QAAE,iC;O;K;IAL9B,iC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;wEAGX,yB;MAAA,6C;  
MAAA,oC;MAAA,gC;MAAA,uC;QAUoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,C  
AAC,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;QACTd,OAAO,I;O;KAXX,C;IAcA,2B;MAMI,OAAO,EC1wB  
yC,qBAAU,CD0wBnD,C;K;wEAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QA  
AhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;  
O;KAPX,C;4EUAU,qB;MAKI,OAAO,gB;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAKoB,Q  
;QADhB,YAAY,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAA  
wB,qB;;QAC9C,OAAO,K;O;KANX,C;0EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QAD  
hB,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,UAAU,WAAV,EAAuB,oBAAV  
B,C;;QACpC,OAAO,W;O;KAXX,C;wFAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAYoB,UAA8B,  
M;QAF9C,YAAY,C;QACZ,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,WAAU,  
cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,oBAAhC,C;;QACpC,OAAO,W;O;KAbX,C;mFAgBA,yB;MAA  
A,uD;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,cAAhB,C;  
UACI,cAAc,UAAU,kCAAI,YAAJ,EAAI,oBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAdX,C;iGai  
BA,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAUI,YAAY,wB;QACZ,kBAAkB,O;QACIB,OAAO,SAAS,cAAhB,C;  
UACI,cAAc,UAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAh  
BX,C;gFamBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,  
C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;O;KAJ1B,C;8FAOA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;  
QAOiB,UAAa,M;QAD1B,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sB

AAP,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,2B;MAWiB,Q;MAFb,ICp4BgD,qBAAU,CD04B1D,C;QAAe,MAA  
M,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QA  
CR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;4EAGX,yB;MAAA,sE;MAAA,uD;MAAA,oC  
;MAAA,sC;QAWI,ICx5BgD,qBAAU,CDw5B1D,C;UAAe,MAAM,6B;QACrB,cAAc,qBAAK,CAAL,C;QACd,gB  
AAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,a  
AAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAA  
W,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;wFA2BA,yB;MAAA,uD;  
MAAA,oC;MAAA,sC;QAOL,IC/6BgD,qBAAU,CD+6B1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QA  
Cd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;Q  
ACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2  
BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAA  
A,sE;MAAA,oC;MAAA,uD;Md/pCA,iB;Mc+pCA,sC;QAEiB,Q;QAFb,IC58BgD,qBAAU,CD48B1D,C;UAAe,MA  
AM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC  
AAK,CAAL,EAAT,C;UACR,WdxqCG,MAAO,KcwqCO,QdxqCP,ECwqCiB,CdxqCjB,C;;Qc0qCd,OAAO,Q;O;K  
AnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdhsCA,iB;McgsCA,sC;QAEiB,Q;QAFb,ICl+Bgd,qBA  
AU,CDk+B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA  
V,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdzcCG,MAAO,KcysCO,QdzcCP,EcysCiB,CdzcCjB,  
C;;Qc2sCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,ICt  
/BgD,qBAAU,CDs/B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aA  
AU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAA  
W,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;MdjuCA,iB;MciuCA,sC;QAaiB,Q;QA  
Fb,IC5gCgD,qBAAU,CD4gC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAA  
b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdxuCG,MAAO,KcwuCO,QdxuCP,ECw  
uCiB,CdxuCjB,C;;Qc0uCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MdhwCA,iB;McgwCA,sC;  
QAaiB,Q;QAFb,ICiCgD,qBAAU,CDgiC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QA  
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdvwCG,MAAO,KcuwCO,  
QdvwCP,ECuwCiB,CdvwCjB,C;;QcywCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;  
QAWiB,Q;QAFb,ICljCgD,qBAAU,CDkjC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;Q  
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KA  
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,k  
D;QAaiB,Q;QAFb,ICxkCgD,qBAAU,CDwkC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT  
,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,  
QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,y  
B;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,IC5lCgD,qBAAU,CD4lC1D,C;UAAe,OAAO,I;QACtB,eA  
Ae,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT  
,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA  
AO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,IC5mCgD,qBAAU,CD4mC1D,C;QAAe,OAAO,I;MACtB,UAAU  
,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CA  
AV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWiB,Q;MAFb,IC9nCgD,qBAAU,CD8nC1D,C;QAAe  
,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,  
C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA  
O,G;K;IAGX,iD;MAOiB,Q;MAFb,IC5oCgD,qBAAU,CD4oC1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAA  
L,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EA  
Aa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAWiB,Q;MAFb,IC9pC  
gD,qBAAU,CD8pC1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,i  
B;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;4EAGX  
,yB;MAAA,sE;MAAA,uD;MAAA,oC;MAAA,sC;QAWI,IClrCgD,qBAAU,CDkrC1D,C;UAAe,MAAM,6B;QACr  
B,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QA

C3B,eAAe,SAAS,oBAAT,C;QACf,AAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;wFA2BA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICzsCgD,qBAAU,CDysC1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,AAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdruCA,iB;McquCA,sC;QAeiB,Q;QAFb,ICtugD,qBAAU,CDsuC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd9uCG,MAAO,Kc8uCO,Qd9uCP,Ec8uCiB,Cd9uCjB,C;;QcgvCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdtwCA,iB;McsWCA,sC;QAeiB,Q;QAFb,IC5vCgD,qBAAU,CD4vC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd/wCG,MAAO,Kc+wCO,Qd/wCP,Ec+wCiB,Cd/wCjB,C;;QcixCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,IChxCgD,qBAAU,CDgxC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;MdvYCA,iB;McuYCA,sC;QAaiB,Q;QAFb,ICtyCgD,qBAAU,CDsyC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd9yCG,MAAO,Kc8yCO,Qd9yCP,Ec8yCiB,Cd9yCjB,C;;QcgzCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;Mdt0CA,iB;Mcs0CA,sC;QAaiB,Q;QAFb,IC1zCgD,qBAAU,CD0zC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd70CG,MAAO,Kc60CO,Qd70CP,Ec60CiB,Cd70CjB,C;;Qc+0Cd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,IC50CgD,qBAAU,CD40C1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,ICl2CgD,qBAAU,CDk2C1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAT,C,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICt3CgD,qBAAU,CDs3C1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,AAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAT,C,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICt4CgD,qBAAU,CDs4C1D,C;QAAe,OAAO,I;MACTB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,AAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWiB,Q;MAFb,ICx5CgD,qBAAU,CDw5C1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,AAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAOiB,Q;MAFb,ICt6CgD,qBAAU,CDs6C1D,C;QAAe,OAAO,I;MACTB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,AAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,OCr7CgD,qBAAU,C;K;0EDw7C9D,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;;QACrD,OAAO,I;O;KAPX,C;8EAUa,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAKmC,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;QAARc,gB;O;KALJ,C;4FAQA,yB;MAAA,6B;MAAA,sC;MAZlBA,6C;MAAA,oC;MAAA,gC;MAylBA,2BAQiB,yB;QAjmBjB,6C;QAAA,oC;QAAA,gC;eAimBiB,0B;UAAA,4B;YAAE,aAAe,c;YA1lBjB,gB;YADb,YAAY,C;YACC,0B;YAAb,OAAa,cAAb,C;cAAa,iC;cAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;YA0lBmB,W;W;S;OAAzB,C;MARjB,oC;QA1lBiB,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA0lBnB,gB;O;KARJ,C;8EAWA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICl+

CgD,qBAAU,CDk+C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KAnBX,C;4FAsBA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICx/CgD,qBAAU,CDw/C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAaV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;wGAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC9gDgD,qBAAU,CD8gD1D,C;UACI,OAAO,I;QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAaV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;0FAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAIbqB,Q;QAHjB,ICriDgD,qBAAU,CDqiD1D,C;UACI,OAAO,I;QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KApBX,C;uFAuBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCAA9B,C;QACrB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,kCAAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O;KAnBX,C;qGAsBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCAA9B,C;QACrB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAaV,EAAiB,iCAAI,KAaJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;iHAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAaV,EAAiB,iCAAI,KAaJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;mGAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,kCAAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O;KApBX,C;wFAuBA,yB;MAAA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAgBoB,Q;QAHhB,ICtpDgD,qBAAU,CDspD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAazB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,af32DO,W;Qe42DP,kBAakB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;sGAuBA,yB;MAAA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAAA,gD;QAIbKB,gC;QAHd,IC9qDgD,qBAAU,CD8qD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAazB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,afn4DO,W;Qe04DP,kBAakB,O;QACJ,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAA0D;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4FAwBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,Q;QAHtB,ICrsDgD,qBAAU,CDqsD1D,C;UAAe,OAAO,W;QACtB,sBAakB,qBAAK,CAAL,CAAIb,C;QACqC,kBAaxB,eAAgB,gBAAhB,C;QAAgC,sBAAI,0BAAJ,C;QAA7C,af35DO,W;Qe45De,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,0BAAV,EAAuB,iCAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KApBX,C;0GAuBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAIb0B,Q;QAHtB,IC7tDgD,qBAAU,CD6tD1D,C;UAAe,OAAO,W;QACtB,sBAakB,qBAAK,CAAL,CAAIb,C;QACqC,kBAaxB,eAAgB,gBAAhB,C;QAAgC,sBAAI,0BAAJ,C;QAA7C,afn7DO,W;Qe07De,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,KAaV,EAAiB,0BAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0EAWBA,yB;MA9FA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MA8FA,gD;QAcW,sB;;UA5FS,Q;UAHhB,ICtpDgD,qBAAU,CDspD1D,C;YAAe,qBAAO,OA+FH,OA/FG,C;YAAP,uB;;UACuB,kBAazB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBA8F3B,OA9F2B,C;UAA9C,af32DO,W;Ue42DP,kBA6FmB,O;UA5FH,0B;UAAhB,OAAgB,cAAhB,C;YAAgB,oC;YACZ,cA2FwB,SA3FV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;QAwFP,yB;O;KADJ,C;wFAiBA,yB;MAxFA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAwFA,gD;QAEw,6B;;UAtFO,gC;UAHd,IC9qDgD,qBAAU,CD8qD1D,C;YAAe,4BAAO,OAYFI,OAZFJ,C;YAAP,8B;;UACuB,kBAazB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBAwFpB,OAXFoB,C;UAA9C,afn4DO,W;Ue04DP,kBAuF0B,O;UAtFZ,6B;UAAA,mB;UAAA,kB;UAAA,kB;UAA0D;YACI,cAqF+B,SArFjB,CAAU,KAaV,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;QAKFP,gC;O;

KAFJ,C;4EAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;wFAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACF,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAVX,C;4EAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACJ,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,SASoB,gB;MATpB,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACgB,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,cAAO,SAAS,oBAAT,CAAP,C;;QAEX,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M9BzuDA,6B;M8ByuDA,sC;QAWoB,Q;QADhB,U9BzuDmC,c8ByuDnB,C9BzuDmB,C;Q8B0uDnB,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,M9BvjEiD,c8BujEjD,G9BvjE2D,KAAK,G8BujEzD,SAAS,oBAAT,C9BvjEoE,KAAAX,IAAf,C;;Q8ByjErD,OAAO,G;O;KAdX,C;4EAiBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MbvDA,+B;MauvDA,sC;QAWoB,Q;QADhB,UbtvDqC,eAAW,oBasvD/B,CbtvD+B,CAAX,C;QauvDrB,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,MbrkEmD,eaqkEnD,GbrkE8D,KAAK,KaqkE5D,SAAS,oBAAT,CbrkEuE,KAAAX,CAAhB,C;;QaukEvD,OAAO,G;O;KAdX,C;IAiBA,oC;MAWI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,+C;MAGBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,EAAwD,SAAXD,C;K;IAcsB,oC;MAAE,OAAA,EAAG,W;K;IAXtC,0C;MAWI,OAAO,6BAAgB,IAAhB,EAAcB,sBAAtB,C;K;IAGX,uD;MAGBI,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAAsB,IAA9C,EAAG,EAAhE,C;K;oFAGX,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;QAGf,OAAO,cAAK,KAAL,EAAy,MAAZ,C;O;KAjBX,C;oFAoBA,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;QAGf,OAAO,cAAK,KAAM,WAAAX,EAAuB,MAAO,WAA9B,C;O;KAjBX,C;IAqCgD,6B;MAAE,OAAA,EAAG,W;K;IAjBrD,2D;MAGB4C,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACjF,OAAO,sBAAS,IAAT,EAAe,IAAf,EAaqB,cAArB,EAAqC,eAArC,C;K;IAGX,sE;MAkBgD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAQhE,Q;MAPrB,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,eAAe,SAAK,O;MACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;MACrB,aAAa,iBAAa,cAAb,C;MACb,YAAY,C;MACZ,OAAGB,CAAT,qBAAiB,QAAxB,C;QACI,UAAU,QAAQ,IAAR,I;QACO,IAAI,MAAM,CAAN,IAAW,MAAM,QAArB,C;UAAiC,IAAI,cAAJ,C;YAAoB,e;;YAAc,K;;UAAa,U;QAAjG,qB;QACA,MAAO,WAAI,UAAU,8BAAy,KAAZ,EAAmB,UAAAnB,CAAV,CAAJ,C;QACP,gBAAS,IAAT,I;;MAEJ,OAAO,M;K;IAoB6C,qC;MAAE,OAAA,EAAG,W;K;IAjB7D,iE;MAGBoD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACzF,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA6C,uBAA7C,C;K;IAwByB,2F;MAAA,wB;QAC5B,UAAU,QAAQ,YAAR,I;QACV,iBAAqB,MAAM,CAAN,IAAW,MAAM,4BAArB,GAA6B,4BAA7B,GAAyC,G;QAD1D,OAEA,kBAAU,0CAAY,KAAZ,EAAmB,UAAAnB,CAAV,C;O;K;IAxBR,gF;MAkBWd,sB;QAAA,SAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAC7F,oBAAoB,IAApB,EAA0B,MAA1B,C;MACA,cAAc,KAAK,cAAJ,GAAoB,yBAApB,GAAiC,WAAQ,mBAAS,IAAT,GAAgB,CAAhB,IAAR,CAAIC,EAAkE,MAAIE,C;MACd,OAA4B,OAAb,aAAR,OAAQ,CAAa,EAAI,qDAAJ,C;K;IAOhC,kC;MAkBI,ad1nEO,MAAO,Kc0nEU,gBd1nEV,Ec+mEH,KAW2B,Od1nExB,C;Mc2nEd,WAAW,iBAAa,MAAb,C;MACX,aAAU,CAAV,MAAkB,MAAIB,M;QACI,IAAK,WAdqB,GAcP,iCAAK,CAAL,EAdO,EAcE,YAdrB,KAcqB,YAAM,CAAN,EAdF,CACrB,C;;MAdT,OAGBo,I;K;wEAbX,yB;MAAA,gE;MAAA,oC;MdxnEA,iB;McwnEA,8C;QAQI,ad1nEO,MAAO,Kc0nEK,SAAK,Od1nEV,Ec0nEkB,KAAAM,Od1nExB,C;Qc2nEd,WAAW,eAAa,MAAb,C;QACX,aAAU,CAAV,MAAkB,MAAIB,M;UACI,IAAK,WAAI,UAAU,iCAAK,CAAL,EAAV,EAAmB,6BAAM,CAAN,EAAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAGBA,kC;MASW,sB;;QAaP,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,qBAAO,W;UAAP,uB;;QACd,aAAa,iBAAa,IAAb,C;QACb,iBAAc,CAAd,UAAAsB,IAATB,U;UACI,MAAO,WAjBkB,GAiBJ,iCAAK,KAAL,EAjBI,EAiBS,iCAAK,QAAQ,CAAR,IAAL,EAjBT,CAiBIB,C;;QAEX,qBAAO,M;;MAnBP,yB;K;uFAGJ,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,uC;QAUI,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,

OAAO,W;QACrB,aAAa,eAAa,IAAb,C;QACb,iBAAC,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WAAI,UAAU,iCA  
AK,KAAL,EAAV,EAAuB,iCAAK,QAAQ,CAAR,IAAL,EAAvB,CAAJ,C;;QAEX,OAAO,M;O;KAhBX,C;IAwBo  
B,8C;MAAA,mB;QAAE,OAAK,WAAAL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OCloE0B,qBAA  
U,C;;MDkoE1D,S;QAAiC,OAAO,W;MACxC,oCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,WAA  
AL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OCloE0B,qBAAU,C;;MD0oE1D,S;QAAiC,OAAO,e;  
MACxC,oCAAgB,8BAAhB,C;K;IEp2EkC,yC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCiMV,K;O;K;IkCjMH  
,wC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCyOV,K;O;K;IkCzOC,yC;MAAA,wB;QAAW,OAAA,aAAK,K  
AAL,CjBoPV,K;O;K;IkBpPC,0C;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CiC4MV,K;O;K;4FmCtQzC,qB;MA  
UI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CA  
AJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;M  
AUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,C  
AAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;  
MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,  
CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,q  
B;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBA  
AI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;uGAuC  
X,yB;MAkhHI,8D;MAlhHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAygHT,cAAR,iBAAQ,C;;QAzgHhB,OAAO,  
OAAcS,sBAAI,KAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYAY,yB;MA8gHI,8D;MA9gHJ,iD;QASe,o  
BAAS,C;QAAT,S;UAAc,gBAqgHT,cAAR,iBAAQ,C;;QArgHhB,OAAO,OAAcS,sBAAI,KAJ,CAAtC,GAAsD,a  
AAa,KAAb,C;O;KATjE,C;uGAYAY,yB;MA0gHI,8D;MA1gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAigHT,cA  
AR,iBAAQ,C;;QAJgHhB,OAAO,OAAcS,sBAAI,KAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYAY,yB;  
MAsgHI,8D;MATgHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBA6/GT,cAAR,iBAAQ,C;;QA7/GhB,OAAO,OAAcS,  
sBAAI,KAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYAY,yB;MAAA,sD;MAAA,mC;QASI,OAAAY,UA  
AL,SAAK,EAAU,KAUV,C;O;KAThB,C;uGAYAY,yB;MAAA,sD;MAAA,mC;QASI,OAAAY,UAAL,SAAK,EAAU,  
KAUV,C;O;KAThB,C;uGAYAY,yB;MAAA,sD;MAAA,mC;QASI,OAAAY,UAAL,SAAK,EAAU,KAUV,C;O;KATh  
B,C;uGAYAY,yB;MAAA,sD;MAAA,mC;QASI,OAAAY,UAAL,SAAK,EAAU,KAUV,C;O;KAThB,C;iFAYA,gC;M  
ASW,sB;;QAkOS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAIOH,SAkOO,CAAU,OAAV,C  
AAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAnOP,yB;K;iFAGJ,gC;MASW,sB;;;QAIOS,Q;QAA  
A,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAjOH,SAiOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;  
YAAP,uB;;;QAC9C,qBAAO,I;;;MAIOP,yB;K;iFAGJ,gC;MASW,sB;;;QA+NS,Q;QAAA,2B;QAAhB,OAAgB,cAA  
hB,C;UAAgB,yB;UAAM,IAhOH,SAgOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBA  
AO,I;;;MAjOP,yB;K;iFAGJ,gC;MASW,sB;;;QA+NS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM  
,IA/NH,SA+NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAhOP,yB;K;yFA  
GJ,yB;MAgoBA,+C;MAkuFI,0D;MAI2GJ,uC;QASW,qB;;;UAgoBO,Q;UAAA,OAAa,SAytFX,YAAR,iBAAQ,CA  
ztFW,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAkoBV,CAAU,  
OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QApoBP,wB;O;KATJ,C;yFAYA,yB;MAooB  
A,+C;MA0tFI,0D;MA91GJ,uC;QASW,qB;;;UAooBO,Q;UAAA,OAAa,SAitFX,YAAR,iBAAQ,CAjtFW,CAAb,W;  
UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAtoBc,SAsoBV,CAAU,OAAV,CAAJ,C;  
cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAxoBP,wB;O;KATJ,C;yFAYA,yB;MAwoBA,+C;MAktFI,0D  
;MA11GJ,uC;QASW,qB;;;UAwoBO,Q;UAAA,OAAa,SAysFX,YAAR,iBAAQ,CAzsFW,CAAb,W;UAAAd,OAAC,c  
AAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAoBV,CAAU,OAAV,CAAJ,C;cAAwB,oBA  
AO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA5oBP,wB;O;KATJ,C;yFAYA,yB;MA4oBA,+C;MA0sFI,0D;MA11GJ,uC;  
QASW,qB;;;UA4oBO,Q;UAAA,OAAa,SAisFX,YAAR,iBAAQ,CAjsFW,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,u  
B;YACV,cAAc,sBAAK,KAAL,C;YACd,IA9oBc,SA8oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;  
;UAE5B,oBAAO,I;;;QAhpBP,wB;O;KATJ,C;mFAYA,yB;MAAA,8C;MpC1GA,6B;MoC0GA,4B;QASI,OpCzGm  
C,coCyGpB,MAAR,iBAAQ,CpCzGoB,C;O;KoCgGvC,C;mFAYA,yB;MAAA,8C;MnBvGA,+B;MmBuGA,4B;QA  
SI,OnBtGsC,emBsGvB,MAAR,iBAAQ,CnBtGuB,C;O;KmB6F1C,C;mFAYA,yB;MAAA,8C;MrChLA,+B;MqCgL  
A,4B;QASI,OrC/KsC,eqC+KvB,MAAR,iBAAQ,CrC/KuB,C;O;KqCsK1C,C;mFAYA,yB;MAAA,8C;MnC/KA,iC;



c,uB;YACV,IuCp5ByB,UpC3bM,cH+0CjB,YAAK,KAAL,CG/0CiB,CoC2bN,CvCo5BzB,C;cACI,qBAAO,K;cAA  
P,uB;;;UAGR,qBAAO,E;;;QuCx5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvCw5BA,0D;MAAA,+C;MoBj1CA,  
+B;MmBybA,yBAO+B,yB;QnBhc/B,+B;emBgc+B,6B;UAAA,qB;YAAE,yBnBtbS,emBsbC,EnBtbD,CmBsbT,C;  
W;S;OAAF,C;MAP/B,uC;QAOMB,kBAAR,iB;QAAQ,sB;;UvCq5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W  
;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IuCt5ByB,UnBtbS,epB40CpB,YAAK,KAAL,CoB50CoB,CmBsbT,CvCs  
5BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC15BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvC02  
BA,0D;MAAA,+C;ME12CA,+B;MqCggBA,yBAO+B,yB;QrCvgB/B,+B;eqCugB+B,6B;UAAA,qB;YAAE,yBrC7f  
S,eqC6fC,ErC7fD,CqC6fT,C;W;S;OAAF,C;MAP/B,uC;QAOMB,kBAAR,iB;QAAQ,sB;;UvCu2BD,Q;UAAA,OA  
AQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IuCx2ByB,UrC7fS,eFq2CpB,YAAK,KAA  
L,CER2CoB,CqC6fT,CvCw2BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC52BP,yB;O;KAPJ,C;+FA  
UA,yB;MAAA,sC;MvC42BA,0D;MAAA,+C;MIz2CA,iC;MmC6fA,yBAO+B,yB;QnCpgB/B,iC;emCogB+B,6B;U  
AAA,qB;YAAE,yBnC1fY,gBmC0fF,EnC1fE,CmC0fZ,C;W;S;OAAF,C;MAP/B,uC;QAOMB,kBAAR,iB;QAAQ,s  
B;;UvCy2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IuC12ByB,Un  
C1fY,gBjO2CvB,YAAK,KAAL,CIp2CuB,CmC0fZ,CvC02BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;  
QuC92BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,4C;MpCteA,6B;MoCseA,4B;QAWI,OpCvemC,coCuepB,KAAR,iBA  
AQ,CpCveoB,C;O;KoC4dvC,C;iFACa,yB;MAAA,4C;MnBreA,+B;MmBqeA,4B;QAWI,OnBtesC,emBsevB,KAA  
R,iBAAQ,CnBteuB,C;O;KmB2d1C,C;iFACa,yB;MAAA,4C;MrChjBA,+B;MqCgjBA,4B;QAWI,OrCjjBsC,eqCijB  
vB,KAAR,iBAAQ,CrCjjBuB,C;O;KqCsiB1C,C;iFACa,yB;MAAA,4C;MnCjjBA,iC;MmCijBA,4B;QAWI,OnCljB  
yC,gBmCkjB1B,KAAR,iBAAQ,CnCljB0B,C;O;KmCuiB7C,C;iFACa,yB;MAAA,+C;MAAA,iE;MA83FI,0D;MA9  
3FJ,uC;QAWkB,Q;QAAA,OAAa,SAm3FX,YAn3FF,SAm3FN,QAAQ,CAn3FW,CAAb,W;QAAd,OAAc,cAAAd,C;  
UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAA  
M,gCAAUb,mDAAvB,C;O;KafV,C;iFakBA,yB;MAAA,+C;MAAA,iE;MAo3FI,0D;MAP3FJ,uC;QAWkB,Q;QA  
AA,OAAa,SAY2FX,YAZ2FF,SAY2FN,QAAQ,CAZ2FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,  
sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAUb,mDAAvB,  
C;O;KafV,C;iFakBA,yB;MAAA,+C;MAAA,iE;MA02FI,0D;MA12FJ,uC;QAWkB,Q;QAAA,OAAa,SA+1FX,YA  
/1FF,SA+1FN,QAAQ,CA/1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,  
IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAUb,mDAAvB,C;O;KafV,C;iFakBA,yB;  
MAAA,+C;MAAA,iE;MAg2FI,0D;MAh2FJ,uC;QAWkB,Q;QAAA,OAAa,SAq1FX,YAr1FF,SAq1FN,QAAQ,CAr  
1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAA  
J,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAUb,mDAAvB,C;O;KafV,C;+FakBA,yB;MAAA,0D;MAAA,qC;Q  
AOI,OAAe,YAAR,iBAAQ,EAAY,OpCxsBM,KoCwsBIB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QA  
OI,OAAe,YAAR,iBAAQ,EAAY,OnBvsBQ,KmBusBpB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,  
OAAe,YAAR,iBAAQ,EAAY,OrCpwBQ,KqCowBpB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,O  
AAe,YAAR,iBAAQ,EAAY,OnCnwBU,KmCmwBtB,C;O;KAPnB,C;IAUA,kC;MAQI,OAAW,mBAAJ,GAAe,IA  
Af,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,i  
BAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,  
C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;6FAGpC,yB;MA  
AA,+C;MAkuFI,0D;MALuFJ,uC;QASkB,Q;QAAA,OAAa,SAytFX,YAZtFF,SAytFN,QAAQ,CAZtFW,CAAb,W;Q  
AAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA  
AO,O;;QAEEnC,OAAO,I;O;KAbX,C;6FAGBA,yB;MAAA,+C;MA0tFI,0D;MA1tFJ,uC;QASkB,Q;QAAA,OAAa,S  
AitFX,YAjtFF,SAitFN,QAAQ,CAjtFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,  
C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,OAAO,I;O;KAbX,C;6FAGBA,yB;MAAA,+C;  
MAktFI,0D;MAItFJ,uC;QASkB,Q;QAAA,OAAa,SAYSFX,YAZsFF,SAYSFN,QAAQ,CAZsFW,CAAb,W;QAAd,O  
AAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;  
QAEEnC,OAAO,I;O;KAbX,C;6FAGBA,yB;MAAA,+C;MA0sFI,0D;MA1sFJ,uC;QASkB,Q;QAAA,OAAa,SAisFX,  
YAjsFF,SAisFN,QAAQ,CAjsFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UA  
Cd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,OAAO,I;O;KAbX,C;qFAGBA,yB;MAAA,mC;MAA  
A,gD;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAS





DI,IAAI,EwBijDI,KAAK,CxBjjDT,CAAJ,C;QACI,cwBgiDc,sD;QxB/iDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBgiDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBjDI,IAAI,EwB+jDI,KAAK,CxB/jDT,CAAJ,C;QACI,cwB8jDc,sD;QxB7jDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB8jDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBnkDI,IAAI,EwB6kDI,KAAK,CxB7kDT,CAAJ,C;QACI,cwB4kDc,sD;QxB3kDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB4kDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,mC;MxBjDI,IAAI,EwB2IDI,KAAK,CxB3IDT,CAAJ,C;QACI,cwB0IDc,sD;QxBzIDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB0IDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxB/IDI,IAAI,EwBymDI,KAAK,CxBzmDT,CAAJ,C;QACI,cwBwmDc,sD;QxBvmDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBwmDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxB7mDI,IAAI,EwBunDI,KAAK,CxBvnDT,CAAJ,C;QACI,cwBsnDc,sD;QxBrnDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBsnDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxB3nDI,IAAI,EwBqoDI,KAAK,CxBroDT,CAAJ,C;QACI,cwBooDc,sD;QxBnoDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBooDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;mGAGX,yB;MAAA,4C;MAAA,qD;MAkqEI,8D;MAlqEJ,uC;QASI,iBAypEgB,cAAR,iBAAQ,CAzpEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGaiBA,yB;MAAA,4C;MAAA,qD;MAypEI,8D;MAzpEJ,uC;QASI,iBAgpEgB,cAAR,iBAAQ,CAhpEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGaiBA,yB;MAAA,4C;MAAA,qD;MAhpEI,8D;MAhpEJ,uC;QASI,iBAuoEgB,cAAR,iBAAQ,CAvoEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGaiBA,yB;MAAA,4C;MAAA,qD;MAuoEI,8D;MAvoEJ,uC;QASI,iBA8nEgB,cAAR,iBAAQ,CA9nEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAALI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAALI,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KAI BX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAALI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAALI,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KAI BX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAALI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAALI,IAAJ,C;YACL,WAAW,I;; ;QAE nB,OAAO,I;O;KAI BX,C;qFAqBA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAgRA,Q;QAAA,2B;QA AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAhRa,SAgRT,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAALI,OAA J,C;;QAhR1D,OAIRO,W;O;KA1RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAIrA,Q;QAAA,2 B;QA AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAjRc,SAiRV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAALI, OAAJ,C;;QAJR1D,OAKRO,W;O;KA3RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAKRA,Q;Q AAA,2B;QA AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAIRc,SAkRV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY ,WAALI,OAAJ,C;;QAIR1D,OAmRO,W;O;KA5RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAm RA,Q;QAAA,2B;QA AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA nRe,SAmRX,CAAU,OAAV,CAAJ,C;YAAw B,WAAY,WAALI,OAAJ,C;;QAnR1D,OAO RO,W;O;KA7RX,C;kGAYA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAm6HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA 12HT,IAzDsC,SAyDIC,EA 02HkB,cA12HIB,EA02HkB,sBA12HIB,WA02H2B,IA12H3B,CAAJ,C;YAA2C,sBA02HZ,IA12HY,C;;QAZD/C,O A2DO,W;O;KATeX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAK6HV,gB;QADb,YAAY,C;Q ACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA t2HT,IA5DuC,SA4DnC,EAs2HkB,cAt2HIB,EAs2HkB,sBA t2HIB,W As2H2B,IA t2H3B,CAAJ,C;YAA2C,sBA s2HZ,IA t2HY,C;;QA5D/C,OA8DO,W;O;KAZeX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAI6HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;U

AI2HT,IA/DuC,SA+DnC,Eak2HkB,cAI2HIB,Eak2HkB,sBAI2HIB,Wak2H2B,IAI2H3B,CAAJ,C;YAA2C,sBAk2  
HZ,IAI2HY,C;;QA/D/C,OAIeO,W;O;KA5EX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAg6  
HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA91HT,IAIEwC,SAkEpC,EA81HkB,cA91  
HIB,EA81HkB,sBA91HIB,WA81H2B,IA91H3B,CAAJ,C;YAA2C,sBA81HZ,IA91HY,C;;QAIE/C,OAOeO,W;O;K  
A/EX,C;uGAcA,6C;MA3HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QA12HT,IAAI,  
WA02HkB,cA12HIB,EA02HkB,sBA12HIB,WA02H2B,IA12H3B,CAAJ,C;UAA2C,sBA02HZ,IA12HY,C;;MAE/  
C,OAAO,W;K;uGAGX,6C;MAk3HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QA2HT,  
IAAI,WAs2HkB,cAt2HIB,EAs2HkB,sBA2HIB,WAs2H2B,IA2H3B,CAAJ,C;UAA2C,sBAS2HZ,IA2HY,C;;MA  
E/C,OAAO,W;K;uGAGX,6C;MA82HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QA12H  
T,IAAI,Wak2HkB,cAI2HIB,Eak2HkB,sBAI2HIB,Wak2H2B,IAI2H3B,CAAJ,C;UAA2C,sBAk2HZ,IAI2HY,C;;  
MAE/C,OAAO,W;K;uGAGX,6C;MA02HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;Q  
A91HT,IAAI,WA81HkB,cA91HIB,EA81HkB,sBA91HIB,WA81H2B,IA91H3B,CAAJ,C;UAA2C,sBA81HZ,IA91  
HY,C;;MAE/C,OAAO,W;K;2FAGX,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,gB;QAgDH,Q;QAAA,2B;QAAh  
B,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAhDY,SAgDX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,  
OAAJ,C;;QAhd3D,OAI DO,W;O;KA1DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,gB;QAI DH,Q;Q  
AAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAjDa,SAiDZ,CAAU,OAAV,CAAL,C;YAAyB,  
WAAY,WAAI,OAAJ,C;;QAJD3D,OAKDO,W;O;KA3DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,g  
B;QAKDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAIDa,SAkDZ,CAAU,OAAV,CA  
AL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QAID3D,OAmDO,W;O;KA5DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;Q  
ASW,kBAAY,gB;QAmDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAnDc,SAmDb,C  
AAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QAnD3D,OAO DO,W;O;KA7DX,C;+FAYA,6C;MASoB,  
Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WA  
AY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAG  
B,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FA  
GX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL  
,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,c  
AAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,O  
AAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAA  
V,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,O  
AAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,O  
AAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAA  
V,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,O  
AAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,O  
AAO,W;K;IAGX,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OIC3jEe,W;;MkC4jEtC,OAA4D,SA0iDrD,cAAkB,c  
AAR,iBAAQ,EA1iDN,OAAQ,MA0iDF,EA1iDS,OAAQ,aAAR,GAAuB,CAAvB,IA0iDT,CAAI B,CA1iDqD,C;K;I  
AGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OICrKEe,W;;MkCskEtC,OAA4D,SAgjDrD,eAAmB,cAAR,iBA  
AQ,EAhjDP,OAAQ,MAgjDD,EAhjDQ,OAAQ,aAAR,GAAuB,CAAvB,IAgjDR,CAAnB,CAhjDqD,C;K;IAGhE,s  
C;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OIC/kEe,W;;MkCglEtC,OAA4D,UAsjDrD,eAAmB,cAAR,iBAAQ,EA  
tjDP,OAAQ,MAsjDD,EA tjDQ,OAAQ,aAAR,GAAuB,CAAvB,IAsjDR,CAAnB,CA tjDqD,C;K;IAGhE,sC;MAMI,I  
AAI,OAAQ,UAAZ,C;QAAuB,OICzlEe,W;;MkC0lEtC,OAA4D,UA4jDrD,gBAAoB,cAAR,iBAAQ,EA5jDR,OAA  
Q,MA4jDA,EA5jDO,OAAQ,aAAR,GAAuB,CAAvB,IA4jDP,CAApB,CA5jDqD,C;K;IAGhE,sC;MASKB,Q;MAH  
d,WAAmB,wBAAR,OAAQ,EAawB,EAaxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACTB,WAA  
W,iBAAGB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;  
MAET,OAAO,I;K;IAGX,sC;MASKB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAawB,EAaxB,C;MACnB,IAAI,SAA  
Q,CAAZ,C;QAAe,OAAO,W;MACTB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAC,uB;QA  
CV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MASKB,Q;MAHd,WAAmB,wBAAR,OA  
AQ,EAawB,EAaxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACTB,WAAW,iBAAiB,IAAjB,C;MA  
CG,yB;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,s

C;MASKB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAaKB,IAAIB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KA AJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,2C;MAMI,OAAO,cAAkB,aAAR,iBAAQ,EAAW,OAA X,CAAIB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAAIB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAAIB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,aAAR,iBAAQ,EAAW,OAA X,CAA pB,C;K;IAGX,2C;MAMI,OAAO,cAAkB,cAAR,iBAAQ,EAAW,OAA X,CAAIB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,cAAR,iBAAQ,EAAW,OAA X,CAAIB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAA nB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,cAAR,iBAAQ,EAAW,OAA X,CAA pB,C;K;IAGX,+B;MAGBiB,Q;MxBjyEb,IAAI,EwB2xEI,KAAK,CxB3xET,CAAJ,C;QACI,cwB0xEc,sD;QxBzxEd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB0xEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAGB,CAAhB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxBzzEb,IAAI,EwBmzEI,KAAK,CxBnzET,CAAJ,C;QACI,cwBkzEc,sD;QxBjzEd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBkzEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxBj1Eb,IAAI,EwB20EI,KAAK,CxB30ET,CAAJ,C;QACI,cwB00Ec,sD;QxBz0Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB00EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxBz2Eb,IAAI,EwBm2EI,KAAK,CxBn2ET,CAAJ,C;QACI,cwBk2Ec,sD;QxBj2Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBk2EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAaKB,CAAIB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,mC;MxBj3EI,IAAI,EwB23EI,KAAK,CxB33ET,CAAJ,C;QACI,cwB03Ec,sD;QxBz3Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB03EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAGB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxBt4EI,IAAI,EwBg5EI,KAAK,CxBh5ET,CAAJ,C;QACI,cwB+4Ec,sD;QxB94Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB+4EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxB35EI,IAAI,EwBq6EI,KAAK,CxBr6ET,CAAJ,C;QACI,cwBo6Ec,sD;QxBn6Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBo6EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAaKB,CAAIB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;MAGX,yB;MAAA,4C;MAAA,gD;MA2CI,8D;MAT2CJ,uC;QASI,iBA61CgB,cAAR,iBAAQ,CA71ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;QAGf,OAAO,iB;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,gD;MA61CI,8D;MA71CJ,uC;QASI,iBAo1CgB,cAAR,iBAAQ,CAp1ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gB



Al,mBAAJ,C;QA Ae,OAAO,S;MACD,kBA8lBd,eA9lBA,SA8lBW,QvB5/EM,QuB4/EjB,C;MA9lBsB,mB;MAA7  
B,OrBp7FO,W;K;IqBu7FX,mC;MAMI,IAAl,mBAAJ,C;QA Ae,OAAO,S;MACD,kBAgmbd,gBAhmBA,SAgmbY,  
QvB9/EK,QuB8/EjB,C;MAhmBsB,mB;MAA7B,OrB97FO,W;K;IqBi8FX,4C;MAMI,IAAl,mBAAJ,C;QA Ae,OAA  
O,S;MACD,kBAkjBd,cAljBA,SAkjBU,QvBh9EO,QuBg9EjB,C;MALjBsB,8B;MAA7B,OrBx8FO,W;K;IqB28FX,4  
C;MAMI,IAAl,mBAAJ,C;QA Ae,OAAO,S;MACD,kBAojBd,eAAmB,UApjBnB,SAojBW,QAAQ,CAAnB,C;MAP  
jBsB,8B;MAA7B,OrB19FO,W;K;IqBq9FX,4C;MAMI,IAAl,mBAAJ,C;QA Ae,OAAO,S;MACD,kBASjBd,eAtjBA,  
SAsjBW,QvB5/EM,QuB4/EjB,C;MATjBsB,8B;MAA7B,OrB59FO,W;K;IqB+9FX,6C;MAMI,IAAl,mBAAJ,C;QA  
Ae,OAAO,S;MACD,kBAwjBd,gBAxjBA,SAwjBY,QvB9/EK,QuB8/EjB,C;MAXjBsB,8B;MAA7B,OrBt+FO,W;K;  
IqBy+FX,uC;MAQoB,kBAygBT,cAAU,iBvBh9EO,QuBg9EjB,C;MAZgBiB,mB;MAAxB,OAAiC,YrBj/F1B,WqBi  
/F0B,C;K;IAGrC,wC;MAQoB,kBA0gBT,eAAmB,UAAr,iBAAQ,CAAnB,C;MA1gBiB,mB;MAAxB,OAAiC,YrB  
5/F1B,WqB4/F0B,C;K;IAGrC,wC;MAQoB,kBA2gBT,eAAW,iBvB5/EM,QuB4/EjB,C;MA3gBiB,mB;MAAxB,O  
AAiC,YrBvgG1B,WqBugG0B,C;K;IAGrC,wC;MAQoB,kBA4gBT,gBAAY,iBvB9/EK,QuB8/EjB,C;MA5gBiB,m  
B;MAAxB,OAAiC,YrBlhG1B,WqBkhG0B,C;K;4FAGrC,qB;MAQI,OAAO,iB;K;0FAGX,qB;MAQI,OAAO,iB;K;  
4FA+BX,qB;MAQI,OAAO,iB;K;8FAGX,qB;MAQI,OAAO,iB;K;8FAGX,yB;MAAA,yC;MAAA,4B;QAQI,OAA  
O,oBAAW,SAAX,C;O;KARX,C;4FAWA,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,SAAV,C;O;KARX,C  
;8FAWA,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;gGAWA,yB;MAAA,2C;MAAA  
4B;QAQI,OAAO,qBAAY,SAAZ,C;O;KARX,C;IAWA,2C;MASI,OAAy,gBAAL,SAAK,EAAc,KAAc,C;K;IAGh  
B,2C;MASI,OAAy,gBAAL,SAAK,EAAc,KAAc,C;K;IAGhB,2C;MASI,OAAy,gBAAL,SAAK,EAAc,KAAc,C;K  
;IAGhB,2C;MASI,OAAy,gBAAL,SAAK,EAAc,KAAc,C;K;IAGhB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAc,oCA  
Ad,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAc,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,E  
AAc,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAc,oCAAd,C;K;IAGzB,sC;MAQI,OAAy,kBAAL,  
SAAK,C;K;IAGhB,sC;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,s  
C;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,  
gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,  
C;K;IAGzB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAU  
I,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAQW,Q;MAAP,OAAO,  
sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,  
IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,  
EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAyB,  
GAAzB,EAA8B,GAA9B,2BAAsC,M;K;sFAGjD,yB;MvB5hFA,8C;MuB4hFA,kF;QAmB6D,iC;UAAA,oBAAYB,  
C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,c;QvB3hF1H,UuB4hFA,iBvB5hFA,EuB4hFiB,WAA  
Y,QvB5hF7B,EuB4hFsC,iBvB5hFtC,EuB4hFyD,UvB5hFzD,EuB4hFqE,QvB5hFrE,C;QuB6hFA,OAAO,W;O;KA  
rBX,C;wFAwBA,yB;MvB5hFA,8C;MuB4hFA,kF;QAmB+D,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;  
QAAG,wB;UAAA,WAAgB,c;QvB3hF5H,UuB4hFA,iBvB5hFA,EuB4hFiB,WAAy,QvB5hF7B,EuB4hFsC,iBvB5  
hFtC,EuB4hFyD,UvB5hFzD,EuB4hFqE,QvB5hFrE,C;QuB6hFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvB5nFA,  
8C;MuB4nFA,kF;QAmB+D,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,c;Qv  
B3nF5H,UuB4nFA,iBvB5nFA,EuB4nFiB,WAAy,QvB5nF7B,EuB4nFsC,iBvB5nFtC,EuB4nFyD,UvB5nFzD,EuB  
4nFqE,QvB5nFrE,C;QuB6nFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvB5nFA,8C;MuB4nFA,kF;QAmBiE,iC;U  
AAA,oBAAYB,C;QAAG,0B;UAAA,aAakB,C;QAAG,wB;UAAA,WAAgB,c;QvB3nF9H,UuB4nFA,iBvB5nFA,E  
uB4nFiB,WAAy,QvB5nF7B,EuB4nFsC,iBvB5nFtC,EuB4nFyD,UvB5nFzD,EuB4nFqE,QvB5nFrE,C;QuB6nFA,O  
AAO,W;O;KArBX,C;kFAwBA,yB;MAAA,uC;MAAA,4B;QASI,OAAO,mBAAU,iBvBh9EO,QuBg9EjB,C;O;KA  
TX,C;oFAYa,yB;MAAA,gD;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAmB,OAAR,iBAAQ,CAAnB,C;O;KATX,  
C;oFAYa,yB;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAW,iBvB5/EM,QuB4/EjB,C;O;KATX,C;oFAYa,yB;MA  
AA,2C;MAAA,4B;QASI,OAAO,qBAAY,iBvB9/EK,QuB8/EjB,C;O;KATX,C;oFAYa,yB;MAAA,gD;MAAA,uC;  
MAAA,qC;QAWI,OAAO,mBAakB,OAAR,iBAAQ,EAAO,OAAP,CAAIb,C;O;KAXX,C;oFAcA,yB;MAAA,gD;  
MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;  
MAAA,+C;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;  
oFAcA,yB;MAAA,gD;MAAA,2C;MAAA,qC;QAWI,OAAO,qBAoB,OAAR,iBAAQ,EAAO,OAAP,CAApB,C;O

;KAXX,C;4FAcA,yB;MAAA,0D;MAAA,uC;MAAA,gD;QAaI,OAAO,mBAaKB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAaIB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD;QAaI,OAAO,oBAaMB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD;QAaI,OAAO,oBAaMB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C;6FAgBA,yB;MAAA,0D;MAAA,2C;MAAA,gD;QAaI,OAAO,qBAAoB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAApB,C;O;KAbX,C;IAgBA,sD;MAWyC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACHe,OAAR,iBAAQ,EAAK,OpCj8GoB,KoCi8GzB,EAAsB,SAAtB,EAAiC,OAAjC,C;K;IAGZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACIE,OAAR,iBAAQ,EAAK,OnBr8GsB,KmBq8G3B,EAAuB,SAAvB,EAAkC,OAAIC,C;K;IAGZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACIE,OAAR,iBAAQ,EAAK,OrCvgHsB,KqCugH3B,EAAuB,SAAvB,EAAkC,OAAIC,C;K;IAGZ,wD;MAW6C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACpE,OAAR,iBAAQ,EAAK,OnC3gHwB,KmC2gH7B,EAAwB,SAAxB,EAAmC,OAAnc,C;K;8FASr,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;8FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;iFAEJ,yB;MAAA,uC;MvB3oEA,iD;MuB2oEA,qC;QAOqB,4B;QAAA,gBAAU,OpCjxjHM,K;QoCwjHjC,OAAO,mBvB7oEA,2BAxIK,gBAAW,SAAX,EAwIL,CuB6oEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvB7oEA,iD;MuB6oEA,qC;QAOI,OAAO,oBvB/oEA,qBuB+oEW,iBvB/oEX,EAxIK,mBuBuxEgB,OnBvjHO,KJgyCvB,CAwIL,CuB+oEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvB/qEA,iD;MuB+qEA,qC;QAOsB,4B;QAAA,gBAAU,OrCpnHO,K;QqConHnC,OAAO,oBvBjrEA,2BAxIK,eAAy,SAAZ,EAwIL,CuBirEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBjrEA,iD;MuBirEA,qC;QAOuB,4B;QAAA,gBAAU,OnCnnHQ,K;QmCmnHrC,OAAO,qBvBnrEA,2BAxIK,gBAAa,SAAb,EAwIL,CuBmrEA,C;O;KAPX,C;IAUA,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OpCjmHX,K;;MoCkmHjC,OAAO,cAAU,MAAV,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnBlmHT,K;;MmBmmHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnClqHP,K;;MmCmqHrC,OAAO,gBAAY,MAAZ,C;K;iFAGX,yB;MAAA,uC;MvBnuEA,iD;MuBmuEA,sC;QAOI,OAAO,mBvBruEA,qBuBquEU,iBvBruEV,EUbquEoB,QAAS,QvBruE7B,CuBquEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBruEA,iD;MuBquEA,sC;QAOI,OAAO,oBvBvuEA,qBuBuuEW,iBvBvuEX,EUbuuEqB,QAAS,QvBvuE9B,CuBuuEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBvwEA,iD;MuBuwEA,sC;QAOI,OAAO,oBvBzweEA,qBuBywEW,iBvBzweEX,EUBywEqB,QAAS,QvBzweE9B,CuBywEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBzweEA,iD;MuBywEA,sC;QAOI,OAAO,qBvB3wEA,qBuB2weY,iBvB3weZ,EUb2weSb,QAAS,QvB3we/B,CuB2weA,C;O;KAPX,C;IAUA,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAnB,C;K;IAGIB,+C;MAa0B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACzD,oCAAA,2BAaKB,SAaIB,EAA6B,OAA7B,EAAc,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAA,2BAaKB,SAaIB,EAA6B,OAA7B,EAAc,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa4B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC3D,oCAAA,2

BAaKB, SAAIB, EAA6B, OAA7B, EAAsC, cAAtC, C; MACb, YAAU, SAAV, EAAGB, SAAhB, EAA2B, OAA3B, C; K; IAGJ, 0D; MAaI, kBAAK, SAAL, EAAGB, OAAhB, C; MAh8CQ, WAAR, iBAAQ, EAi8CA, SAj8CA, EAi8CW, OAj8CX, C; K; IAo8CZ, 0D; MAaI, kBAAK, SAAL, EAAGB, OAAhB, C; MAj8CQ, WAAR, iBAAQ, EAk8CA, SAi8CA, EAk8CW, OAI8CX, C; K; IAq8CZ, 0D; MAaI, kBAAK, SAAL, EAAGB, OAAhB, C; MAI8CQ, UAAR, iBAAQ, EAm8CA, SAn8CA, EAm8CW, OAn8CX, C; K; IAs8CZ, 0D; MAaI, kBAAK, SAAL, EAAGB, OAAhB, C; MAn8CQ, WAAR, iBAAQ, EAo8CA, SAp8CA, EAo8CW, OAp8CX, C; K; 8FAu8CZ, qB; MAQI, OAAO, iBvB/jGiB, Q; K; 4FuBkG5B, qB; MAQI, OAAO, iBvBtjGiB, Q; K; 8FuByjG5B, yB; MAAA, gD; MAAA, 4B; QAQI, OAAe, OAAAR, iBAAQ, C; O; KARnB, C; gGAWA, qB; MAQI, OAAO, iBvBtlGiB, Q; K; IuB+lGL, gD; MAAA, wB; QAAW, qCAAK, KAAL, C; O; K; IANIC, iC; MAMI, OAAO, iBAAM, cAAN, EAAY, 8BAAZ, C; K; IASY, kD; MAAA, wB; QAAW, qCAAK, KAAL, C; O; K; IANIC, mC; MAMI, OAAO, iBAAM, cAAN, EAAY, gCAAZ, C; K; IASY, kD; MAAA, wB; QAAW, qCAAK, KAAL, C; O; K; IANIC, mC; MAMI, OAAO, iBAAM, cAAN, EAAY, gCAAZ, C; K; IASiB, gD; MAAA, wB; QAAW, yBAAK, KAAL, C; O; K; IANvC, iC; MAMI, OJxqIO, eAAW, +BIwqIA, gBJxqIA, GAAGB, kBIwqIV, 8BJxqIU, CAAhB, CAAX, C; K; gGI2qIX, yB; MAAA, yC; MAAA, 4B; QAQI, OAAO, oBAAW, SvBxpGM, QuBwpGjB, C; O; KARX, C; IAiB2B, 8C; MAAA, wB; QAAW, wBAAK, KAAL, C; O; K; IANtC, gC; MAMI, OH5rIO, cAAU, gCG4rIA, gBH5rIA, GAAe, iBG4rIT, 6BH5rIS, CAaf, CAAV, C; K; 8FG+rIX, yB; MAAA, uC; MAAA, 4B; QAQI, OAAO, mBAAU, SvBxpGO, QuBwpGjB, C; O; KARX, C; IAiB4B, gD; MAAA, wB; QAAW, yBAAK, KAAL, C; O; K; IANvC, iC; MAMI, OFhtIO, eAAW, kBEGtIA, gBFhtIA, EAAGB, kBEGtIV, 8BFhtIU, CAAhB, CAAX, C; K; gGEmtIX, yB; MAAA, gD; MAAA, yC; MAAA, 4B; QAQI, OAAO, oBAAgB, OAAAL, SAAK, CAAhB, C; O; KARX, C; IAiB6B, kD; MAAA, wB; QAAW, 0BAAK, KAAL, C; O; K; IANx, kC; MAMI, ODpuIO, gBAAy, gCCouIA, gBDpuIA, GAAiB, mBCouIX, +BDpuIW, CAAjB, CAAZ, C; K; kGCuuIX, yB; MAAA, 2C; MAAA, 4B; QAQI, OAAO, qBAAy, SvB1sGK, QuB0sGjB, C; O; KARX, C; mGAWA, yB; MAAA, 0D; MAAA, yD; MAAA, uE; MAAA, 2C; QAcI, aAAa, mBAAyC, cAAIB, YAAy, cAAZ, CAaKB, EAaC, EAAd, CAAzC, C; QAsEG, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UArEO, MAsEP, aAAI, OAAJ, EAteE, aAsEF, CAAC, OAAAd, CAAb, C; ; QAtEhB, OAAuB, M; O; KAf3B, C; mGakBA, yB; MAAA, 0D; MAAA, yD; MAAA, uE; MAAA, 2C; QAcI, aAAa, mBAA0C, cAAIB, YAAy, cAAZ, CAaKB, EAaC, EAAd, CAA1C, C; QAsEG, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UArEO, MAsEP, aAAI, OAAJ, EAteE, aAsEF, CAAC, OAAAd, CAAb, C; ; QAtEhB, OAAuB, M; O; KAf3B, C; mGakBA, yB; MAAA, 0D; MAAA, yD; MAAA, uE; MAAA, 2C; QAcI, aAAa, mBAA2C, cAAIB, YAAy, cAAZ, CAaKB, EAaC, EAAd, CAA3C, C; QAsEG, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UArEO, MAsEP, aAAI, OAAJ, EAteE, aAsEF, CAAC, OAAAd, CAAb, C; ; QAtEhB, OAAuB, M; O; KAf3B, C; uGakBA, iD; MAYoB, Q; MAAA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, WAAY, aAAI, OAAJ, EAaC, CAAC, OAAAd, CAAb, C; ; MAEhB, OAAO, W; K; uGAGX, iD; MAYoB, Q; MAAA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, WAAY, aAAI, OAAJ, EAaC, CAAC, OAAAd, CAAb, C; ; MAEhB, OAAO, W; K; uGAGX, iD; MAYoB, Q; MAAA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, WAAY, aAAI, OAAJ, EAaC, CAAC, OAAAd, CAAb, C; ; MAEhB, OAAO, W; K; uFAGX, yB; MAAA, +D; MAoLA, gD; MApLA, uC; QASW, kBAAU, gB; QAKLD, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UACZ, WAnL6B, SAmLIB, CAAU, OAAV, C; UACC, OAAZ, WAAY, EAAO, IAAP, C; ; QApLhB, OAsLO, W; O; KA/LX, C; uFAYA, yB; MAAA, +D; MASLA, gD; MATLA, uC; QASW, kBAAU, gB; QAO LD, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UACZ, WArL6B, SAqLIB, CAAU, OAAV, C; UACC, OAAZ, WAAY, EAAO, IAAP, C; ; QAtLhB, OAwLO, W; O; KAjMX, C; uFAYA, yB; MAAA, +D; MAwLA, gD; MAXLA, uC; QASW, kBAAU, gB; QASLD, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UACZ, WAvL6B, SAuLIB, CAAU, OAAV, C; UACC, OAAZ, WAAY, EAAO, IAAP, C; ; QAxLhB, OA0LO, W; O; KAnMX, C; uFAYA, yB; MAAA, +D; MA0LA, gD; MA1LA, uC; QASW, kBAAU, gB; QAwLD, Q; QAAA, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UACZ, WAzL6B, SAyLIB, CAAU, OAAV, C; UACC, OAAZ, WAAY, EAAO, IAAP, C; ; QA1LhB, OA4LO, W; O; KA rMX, C; qGAYA, yB; MAAA, +D; MA4DA, gD; MA5DA, uC; QAYW, kBAAiB, gB; QA2DR, gB; QADhB, YAAy, C; QACI, 2B; QAAhB, OAAgB, cAAhB, C; UAAGB, yB; UACZ, WA5DoC, SA4DzB, EAAU, cAAV, EAAU, sBAAV, WAAMb, OAAAnB, C; UACC, OAAZ, WAAY, EAAO, IAAP, C; ; QA7DhB, OA+DO,



W;O;KA3EX,C;qGAeA,yB;MAAA,+D;MA+DA,gD;MA/DA,uC;QAYW,kBAaIB,gB;QA8DR,gB;QADhB,YAA  
Y,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WA/DoC,SA+DzB,EAAU,cAAV,EAAU,sBAAV,  
WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAHEhB,OAkEO,W;O;KA9EX,C;qGAeA,yB;MAA  
A,+D;MAkEA,gD;MAIEA,uC;QAYW,kBAaIB,gB;QAiER,gB;QADhB,YAA,Y,C;QACI,2B;QAaHb,OAAGb,cA  
AhB,C;UAAGb,yB;UACZ,WAIEoC,SAkEzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,  
WAAY,EAAO,IAAP,C;;QAnEhB,OAqEO,W;O;KAjFX,C;qGAeA,yB;MAAA,+D;MAqEA,gD;MArEA,uC;QAY  
W,kBAaIB,gB;QAoER,gB;QADhB,YAA,Y,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WArEOC,  
SAqEzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtEhB,OA  
wEO,W;O;KApFX,C;yGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAA,Y,C;QACI,2B;QAaHb,  
OAAGb,cAAhB,C;UAAGb,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ  
,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;Q  
AFzB,YAA,Y,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,  
WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KafX,C;yGakBA,yB;MAAA,g  
D;MAAA,oD;QAWoB,UACS,M;QAFzB,YAA,Y,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WA  
AW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,  
W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAA,Y,C;QACI,2B;QAaHb,OAAG  
B,cAAhB,C;UAAGb,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAnB,C;UACC,OAAZ,WAA  
Y,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KafX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAA  
hB,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WAAW,UAAU,OOAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAE  
hB,OOAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;U  
AAGb,yB;UACZ,WAAW,UAAU,OOAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KAXX,  
C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WAA  
W,UAAU,OOAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD  
;MAAA,oD;QAOoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,WAAW,UAAU,OOAV,C;UAC  
C,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OOAO,W;O;KAXX,C;uFAcA,yB;MAAA,wE;MA4HA,+D;MA5HA,y  
C;QAYW,kBAAU,oB;QA4HD,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,UA7HoD,WA6H1C,  
CAAY,OAAZ,C;UjC59IP,U;UADP,YiC89Ie,WjC99IH,WiC89IwB,GjC99IxB,C;UACL,IAAI,aAAJ,C;YACH,aiC4  
9IuC,gB;YAA5B,WjC39IX,aiC29IgC,GjC39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCw9IA,iB;UACA,IAAK,  
WAAI,OOAJ,C;;QA/HT,OAIIO,W;O;KA7IX,C;uFAeA,yB;MAAA,wE;MAiIA,+D;MAjIA,yC;QAYW,kBAAU,o  
B;QAiID,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,UAlIqD,WAKI3C,CAAY,OAAZ,C;UjCh/I  
P,U;UADP,YiCk/Ie,WjCl/IH,WiCk/IwB,GjCl/IxB,C;UACL,IAAI,aAAJ,C;YACH,aiCg/IuC,gB;YAA5B,WjC/+IX,  
aiC++IgC,GjC/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC4+IA,iB;UACA,IAAK,WAAI,OOAJ,C;;QApIT,OA  
S,MAAT,C;YACA,e;;YAEA,c;;UiC4+IA,iB;UACA,IAAK,WAAI,OOAJ,C;;QAZIT,OA2IO,W;O;KAvJX,C;uFAe  
A,yB;MAAA,wE;MA2IA,+D;MA3IA,yC;QAYW,kBAAU,oB;QA2ID,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;U  
AAGb,yB;UACZ,UA5IsD,WA4I5C,CAAY,OAAZ,C;UjC7iJP,U;UADP,YiC0hJe,WjC1hJH,WiC0hJwB,GjC1hJx  
B,C;UACL,IAAI,aAAJ,C;YACH,aiCwhJuC,gB;YAA5B,WjCvhJX,aiCuhJgC,GjCvhJhC,EAAS,MAAT,C;YACA,e  
;;YAEA,c;;UiCohJA,iB;UACA,IAAK,WAAI,OOAJ,C;;QA9IT,OAGJO,W;O;KA5JX,C;uFAeA,yB;MAAA,wE;MA  
gJA,+D;MAhJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,Uaj  
JiD,WaiJvC,CAAY,OAAZ,C;UjC7iJP,U;UADP,YiC+iJe,WjC/iJH,WiC+iJwB,GjC/iJxB,C;UACL,IAAI,aAAJ,C;Y  
ACH,aiC6iJuC,gB;YAA5B,WjC5iJX,aiC4iJgC,GjC5iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCyiJA,iB;UACA  
,IAAK,WAnJyD,cAmJrD,CAAE,OAaf,CAAJ,C;;QAnJT,OAqJO,W;O;KAIKX,C;uFAGBA,yB;MAAA,wE;MAqJA  
,+D;MArJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAGb,yB;UACZ,UAtJiD,  
WAsJvC,CAAY,OAAZ,C;UjCikJP,U;UADP,YiCokJe,WjCpkJH,WiCokJwB,GjCpkJxB,C;UACL,IAAI,aAAJ,C;Y  
ACH,aiCkkJuC,gB;YAA5B,WjCjkJX,aiCikJgC,GjCjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8jJA,iB;UAC  
A,IAAK,WAxJyD,cAwJrD,CAAE,OAaf,CAAJ,C;;QAxJT,OA0JO,W;O;KAvKX,C;uFAGBA,yB;MAAA,wE;MA0

JA,+D;MA1JA,yD;QAaW,kBAAU,oB;QA0JD,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA3J  
iD,WA2JvC,CAAY,OAAZ,C;UjCvIJP,U;UADP,YiCylJe,WjCzIJH,WiCylJwB,GjCzIJxB,C;UACL,IAAI,aAAJ,C;Y  
ACH,aiCulJuC,gB;YAA5B,WjCtIJX,aiCslJgC,GjCtIJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCmlJA,iB;UACA,  
IAAK,WA7JyD,cA6JrD,CAAe,OAAf,CAAJ,C;;QA7JT,OA+JO,W;O;KA5KX,C;uFAGBA,yB;MAAA,wE;MA+JA  
,+D;MA/JA,yD;QAaW,kBAAU,oB;QA+JD,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAhKiD  
,WAgKvC,CAAY,OAAZ,C;UjC5mJP,U;UADP,YiC8mJe,WjC9mJH,WiC8mJwB,GjC9mJxB,C;UACL,IAAI,aAAJ  
,C;YACH,aiC4mJuC,gB;YAA5B,WjC3mJX,aiC2mJgC,GjC3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCwmJ  
A,iB;UACA,IAAK,WAlKyD,cAkKrD,CAAe,OAAf,CAAJ,C;;QA1KT,OAoKO,W;O;KAjLX,C;2FAGBA,yB;MAA  
A,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;U  
jC59IP,U;UADP,YiC89Ie,WjC99IH,WiC89IwB,GjC99IxB,C;UACL,IAAI,aAAJ,C;YACH,aiC49IuC,gB;YAA5B,  
WjC39IX,aiC29IgC,GjC39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCw9IA,iB;UACA,IAAK,WAAI,OAAJ,C;;  
QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAgB,cAAh  
B,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjCh/IP,U;UADP,YiCk/Ie,WjCl/IH,WiCk/IwB,GjCl/IxB,C;UAC  
L,IAAI,aAAJ,C;YACH,aiCg/IuC,gB;YAA5B,WjC/+IX,aiC++IgC,GjC/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;  
UiC4+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;Q  
AYoB,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjCpgJP,U;UADP,Y  
iCsgJe,WjCtgJH,WiCsgJwB,GjCtgJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCcgJuC,gB;YAA5B,WjCngJX,aiCmgJgC,  
GjCngJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCggJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;K  
AjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UAC  
Z,UAAU,YAAy,OAAZ,C;UjCxhJP,U;UADP,YiC0hJe,WjC1hJH,WiC0hJwB,GjC1hJxB,C;UACL,IAAI,aAAJ,C;  
YACH,aiCwhJuC,gB;YAA5B,WjCvhJX,aiCuhJgC,GjCvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCohJA,iB;U  
ACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA  
,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjC7iJP,U;UADP,YiC+iJe,WjC/iJH,  
WiC+iJwB,GjC/iJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC6iJuC,gB;YAA5B,WjC5iJX,aiC4iJgC,GjC5iJhC,EAAS,M  
AAT,C;YACA,e;;YAEA,c;;UiCyiJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,  
C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAA  
U,YAAy,OAAZ,C;UjCikJP,U;UADP,YiCokJe,WjCpkJH,WiCokJwB,GjCpkJxB,C;UACL,IAAI,aAAJ,C;YACH,ai  
CkkJuC,gB;YAA5B,WjCjkJX,aiCikJgC,GjCjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8jJA,iB;UACA,IAAK,  
WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAA  
A,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjCvIJP,U;UADP,YiCylJe,WjCzIJ  
H,WiCylJwB,GjCzIJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCulJuC,gB;YAA5B,WjCtIJX,aiCslJgC,GjCtIJhC,EAAS,  
MAAT,C;YACA,e;;YAEA,c;;UiCmlJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIB  
X,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA  
AU,YAAy,OAAZ,C;UjC5mJP,U;UADP,YiC8mJe,WjC9mJH,WiC8mJwB,GjC9mJxB,C;UACL,IAAI,aAAJ,C;YA  
CH,aiC4mJuC,gB;YAA5B,WjC3mJX,aiC2mJgC,GjC3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCwmJA,iB;U  
ACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;+EAqBA,yB;MAAA,gE;MAAA,uC;QA  
UW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAu  
Kb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM  
,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IA  
AV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;8EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAA  
b,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,  
C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,  
Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,  
OAwKO,W;O;KAILX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAqDP,gB;QADb,YA  
AY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAtDwB,SAsDpB,EAAU,cAAV,EAAU,sBAA  
V,WAAmB,IAAnB,CAAJ,C;;QAtDhB,OAuDO,W;O;KAjEX,C;6FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM  
,eAAa,cAAb,C;QAwDP,gB;QADb,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WazDw  
B,SAYDpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAzDhB,OA0DO,W;O;KApEX,C;6FAaA,y



CF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,E  
AAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;M  
ACF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,  
EAAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;  
MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAn  
B,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;0FAGX,yB;MA1vDI,8D;MA0vDJ,gD;QAeoC,Q;QAHhC,YAtwDgB,  
cAAR,iBAAQ,C;QAuwDhB,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EA  
AI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MATwDI,8D;MAswDJ,gD;Q  
AeoC,Q;QAHhC,YAlxDgB,cAAR,iBAAQ,C;QAmxDhB,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cA  
Ac,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;  
MAIxDI,8D;MAkxDJ,gD;QAeoC,Q;QAHhC,YA9xDgB,cAAR,iBAAQ,C;QA+xDhB,kBAAkB,O;QACIB,OAAO,  
SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W  
;O;KAjBX,C;2FAoBA,yB;MA9xDI,8D;MA8xDJ,gD;QAeoC,Q;QAHhC,YA1yDgB,cAAR,iBAAQ,C;QA2yDhB,k  
BAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,W  
AAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;yGAoBA,yB;MA10DI,8D;MA00DJ,gD;QAaI,YAv1DgB,cAAR,iBAAQ,  
C;QAw1DhB,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CA  
AjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MAx1DI,8D;MAw1DJ,gD;QAaI,  
YAr2DgB,cAAR,iBAAQ,C;QAs2DhB,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,  
EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MA2DI  
,8D;MA2DJ,gD;QAaI,YAn3DgB,cAAR,iBAAQ,C;QAo3DhB,kBAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UAC  
I,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,  
C;yGAsBA,yB;MAp3DI,8D;MAo3DJ,gD;QAaI,YAj4DgB,cAAR,iBAAQ,C;QAk4DhB,kBAAkB,O;QACIB,OAA  
O,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QA  
EJ,OAAO,W;O;KAnBX,C;uFAsBA,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,O  
AAO,OAAP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OA  
AP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;  
K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;qGAG  
1B,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,E  
AAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,  
cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;M  
AD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAA  
hB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,  
QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;IAGvB,2B;MAyIB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,  
6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA1gEG,gBAAR,iBAAQ,C;MA0gEhB,aAAU,CAAV,iB;QACI,QA  
AQ,sBAAK,CAAL,C;QACR,IpC5xL8D,YoC4xL1D,GpC5xL2E,KAAjB,EoC4xLpD,CpC5xLiF,KAA7B,CoC4xL  
1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAyIB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6  
B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OArhEG,gBAAR,iBAAQ,C;MAqhEhB,aAAU,CAAV,iB;QACI,QAA  
Q,sBAAK,CAAL,C;QACR,InBvyL+D,amBuyL3D,GnBvyL6E,KAAIB,EmBuyLrD,CnBvyLmF,KAA9B,CmBuyL  
3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAyIB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6  
B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAhIEG,gBAAR,iBAAQ,C;MAgiEhB,aAAU,CAAV,iB;QACI,QAA  
Q,sBAAK,CAAL,C;QACR,IrC11L4E,0BqCk1LxE,GrC7IL8B,KAAL,GAAiB,GArP8B,EqCk1LIE,CrC7ILwB,KA  
AL,GAAiB,GArP8B,CqCk1LxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAyIB,Q;MAFb,IAA  
I,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA3iEG,gBAAR,iBAAQ,C;MA2iEhB,a  
AAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InC71L6E,0BmC61LzE,GnC/mL8B,KAAL,GAAiB,KA9  
O+B,EmC61LnE,CnC/mLwB,KAAL,GAAiB,KA9O+B,CmC61LzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;  
K;mFAGX,yB;MAAA,sE;MA1kEI,8D;MA0kEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBA  
AK,CAAL,C;QACd,gBAzIEgB,cAylEA,SAzIER,QAAQ,C;QA0IEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;Q  
AC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ

,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MA/IEI,8D;MA+IEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBA9mEgB,cA8mEA,SA9mER,QAAQ,C;QA+mEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MApEI,8D;MAonEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBA9mEgB,cA8mEA,SA9mER,QAAQ,C;QAooEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MAzoEI,8D;MAyoEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAxpEgB,cAwpEA,SAxpER,QAAQ,C;QAypEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;+FA6BA,yB;MA9rEI,8D;MA8rEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAzsEgB,cAysEA,SAzsER,QAAQ,C;QA0sEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MA/sEI,8D;MA+sEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA1tEgB,cA0tEA,SA1tER,QAAQ,C;QA2tEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAhuEI,8D;MAguEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA3uEgB,cA2uEA,SA3uER,QAAQ,C;QA4uEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAjvEI,8D;MAivEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA5vEgB,cA4vEA,SA5vER,QAAQ,C;QA6vEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MAlyEI,8D;MpBvwHJ,iB;MoByiMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA1zEG,cAAR,iBAAQ,C;QAKzEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnjMG,MAAO,KoBmjMO,QpBnjMP,EoBmjiMiB,CpBnjMjB,C;;QoBqjMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjzEI,8D;MpB/wHJ,iB;MoBgkMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAj0EG,cAAR,iBAAQ,C;QAi0EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1kMG,MAAO,KoB0kMO,QpB1kMP,EoB0kMiB,CpB1kMjB,C;;QoB4kMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAh0EI,8D;MpBvxHJ,iB;MoBulMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA1EG,cAAR,iBAAQ,C;QAg1EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBjmMG,MAAO,KoBimMO,QpBjmMP,EoBimMiB,CpBjmMjB,C;;QoBmmMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/0EI,8D;MpB/xHJ,iB;MoB8mMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/1EG,cAAR,iBAAQ,C;QA+1EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBxnMG,MAAO,KoBwnMO,QpBxnMP,EoBwnMiB,CpBxnMjB,C;;QoB0nMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA93EI,8D;MpBlxHJ,iB;MoBgpMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA94EG,cAAR,iBAAQ,C;QA84EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1pMG,MAAO,KoB0pMO,QpB1pMP,EoB0pMiB,CpB1pMjB,C;;QoB4pMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA74EI,8D;MpB1xHJ,iB;MoBuqMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA75EG,cA

AR,iBAAQ,C;QA65EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBjrMG,MAAO,KoBirMO,QpBjrMP,EoBirMiB,CpBjrMjB,C;;QoBmrMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA55EI,8D;MpBlyHJ,iB;MoB8rMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA56EG,cAAR,iBAAQ,C;QA46EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBxsMG,MAAO,KoBwsMO,QpBxsMP,EoBwsMiB,CpBxsMjB,C;;QoB0sMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA36EI,8D;MpB1yHJ,iB;MoBqtMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA37EG,cAAR,iBAAQ,C;QA27EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/tMG,MAAO,KoB+tMO,QpB/tMP,EoB+tMiB,CpB/tMjB,C;;QoBiuMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA19EI,8D;MA09EJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx+EG,cAAR,iBAAQ,C;QAw+EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAz+EI,8D;MAy+EJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/EG,cAAR,iBAAQ,C;QAU/EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAx/EI,8D;MAw/EJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAAtgFG,cAAR,iBAAQ,C;QAsGfHb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAvgFI,8D;MAugFJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhrFG,cAAR,iBAAQ,C;QAqhFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MATjFI,8D;MpBvwhJ,iB;MoB6zMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApkFG,cAAR,iBAAQ,C;QAokFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBr0MG,MAAO,KoBq0MO,QpBr0MP,EoBq0MiB,CpBr0MjB,C;;QoBu0Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MANkFI,8D;MpB/wHJ,iB;MoBk1MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjIFG,cAAR,iBAAQ,C;QAilFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB11MG,MAAO,KoB01MO,QpB11MP,EoB01MiB,CpB11MjB,C;;QoB41Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAhlFI,8D;MpBvxHJ,iB;MoBu2MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IFG,cAAR,iBAAQ,C;QA8IFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/2MG,MAAO,KoB+2MO,QpB/2MP,EoB+2MiB,CpB/2MjB,C;;QoBi3Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA7IFI,8D;MpB/xHJ,iB;MoB43MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mFG,cAAR,iBAAQ,C;QA2mFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp4MG,MAAO,KoB04MO,QpBp4MP,EoB04MiB,CpBp4MjB,C;;QoB4Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA1oFI,8D;MpBlxHJ,iB;MoB45MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxpFG,cAAR,iBAAQ,C;QAwPfhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp6MG,MAAO,KoB06MO,QpBp6MP,EoB06MiB,CpBp6MjB,C;;QoBs6Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAvpFI,8D;MpB1xHJ,iB;MoBi7MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAqrFG,cAAR,iBAAQ,C;QAqqFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBz7MG,MAAO,KoBy7MO,QpBz7MP,EoBy7MiB,CpBz7MjB,C;;QoB27Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAPqFI,8D;MpBlyHJ,iB;MoBs8MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlrFG,cAAR,iBAAQ,C;QAkrFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB98MG,MAAO,KoB88MO,QpB98MP,EoB88MiB,CpB98MjB,C;;QoBg9Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAjrFI,8D;MpB1yHJ,iB;MoB29MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA+rFG,cAAR,iBAAQ,C;QA+rFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn+MG,MAAO,KoBm+MO,QpBn+MP,EoBm+MiB,CpBn+MjB,C;;QoBq+Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA9tFI,8D;MA8tFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,C

AAT,C;QACF,OA1uFG,cAAR,iBAAQ,C;QA0uFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3uFI,8D;MA2uFI,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAuvFG,cAAR,iBAAQ,C;QAuvFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAxvFI,8D;MAwvFI,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApwFG,cAAR,iBAAQ,C;QAowFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MARwFI,8D;MAqwFI,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjxFG,cAAR,iBAAQ,C;QAixFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MAIzFI,8D;MAkzFJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAh0FG,cAAR,iBAAQ,C;QAg0FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MAj0FI,8D;MAi0FJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/0FG,cAAR,iBAAQ,C;QA+0FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAh1FI,8D;MAg1FJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA91FG,cAAR,iBAAQ,C;QA81FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MA/1FI,8D;MA+1FJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA72FG,cAAR,iBAAQ,C;QA62FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MA94FI,8D;MA84FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA15FG,cAAR,iBAAQ,C;QA05FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sgAqBA,yB;MA35FI,8D;MA25FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAv6FG,cAAR,iBAAQ,C;QAu6FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAx6FI,8D;MAw6FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7FG,cAAR,iBAAQ,C;QAo7FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAR7FI,8D;MAq7FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAj8FG,cAAR,iBAAQ,C;QAi8FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA1+FG,gBAAR,iBAAQ,C;MA0+FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpC5vN8D,YoC4vN1D,GpC5vN2E,KAAjB,EoC4vNpD,CpC5vNiF,KAA7B,CoC4vN1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAj/FG,gBAAR,iBAAQ,C;MAi/FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InBnwN+D,amBmwN3D,GnBnwN6E,KAAIB,EmBmwNrD,CnBnwNmF,KAA9B,CmBmwN3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAx/FG,gBAAR,iBAAQ,C;MAw/FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrC1yN4E,0BqC0yNxE,GrCrjN8B,KAAL,GAAiB,GArP8B,EqC0yNIE,CrCrjNwB,KAAL,GAAiB,GArP8B,CqC0yNxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA//FG,gBAAR,iBAAQ,C;MA+/





AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KA  
1BX,C;mFA6BA,yB;MAAA,sE;MAjzGI,8D;MAizGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,  
sBAAK,CAAL,C;QACd,gBAh0GgB,cAg0GA,SAh0GR,QAAQ,C;QAI0GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAA  
O,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,  
QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAA  
O,O;KA1BX,C;+FA6BA,yB;Mat2GI,8D;MAS2GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sB  
AAK,CAAL,C;QACd,gBAj3GgB,cAi3GA,SAj3GR,QAAQ,C;QAK3GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O  
;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QA  
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,  
O;O;KAtBX,C;+FAyBA,yB;MAv3GI,8D;MAu3GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sB  
AAK,CAAL,C;QACd,gBAI4GgB,cAk4GA,SAI4GR,QAAQ,C;QAm4GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;  
QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QA  
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,  
O;O;KAtBX,C;+FAyBA,yB;MAx4GI,8D;MAw4GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sB  
AAK,CAAL,C;QACd,gBAn5GgB,cAm5GA,SAAn5GR,QAAQ,C;QAO5GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;  
QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QA  
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;  
O;KAtBX,C;+FAyBA,yB;MAz5GI,8D;MAy5GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,  
CAAL,C;QACd,gBAp6GgB,cAo6GA,SAp6GR,QAAQ,C;QAq6GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QA  
C3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QA  
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;  
KAtBX,C;kFAyBA,yB;MAAA,sE;MA18GI,8D;MpBnjHJ,iB;MoB6/NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UA  
Ae,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA19GG,cAAR,iBAAQ,C;QA09GhB,aAA  
U,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBvgOG,MAAO,KoBugOO,QpBvgOP,EoBu  
gOiB,CpBvgOjB,C;;QoBygOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAz9GI,8D;MpB3jHJ,iB;MoBoh  
OA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC  
F,OAz+GG,cAAR,iBAAQ,C;QAY+GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,  
WpB9hOG,MAAO,KoB8hOO,QpB9hOP,EoB8hOiB,CpB9hOjB,C;;QoBgiOd,OAAO,Q;O;KApBX,C;mFAuBA,y  
B;MAAA,sE;MAx+GI,8D;MpBnkHJ,iB;MoB2iOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QAC  
rB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx/GG,cAAR,iBAAQ,C;QAw/GhB,aAAU,CAAV,iB;UACI,Q  
AAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrjOG,MAAO,KoBqjOO,QpBrjOP,EoBqjOiB,CpBrjOjB,C;;QoB  
ujOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAv/GI,8D;MpB3kHJ,iB;MoBkkOA,sC;QAgBiB,Q;QAFb,  
IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAvgHG,cAAR,iBAAQ  
,C;QAugHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB5kOG,MAAO,KoB4kO  
O,QpB5kOP,EoB4kOiB,CpB5kOjB,C;;QoB8kOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAtiHI,8D;Mp  
B9jHJ,iB;MoBomOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CA  
AL,CAAT,C;QACF,OAjtHG,cAAR,iBAAQ,C;QAsjHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C  
AAT,C;UACR,WpB9mOG,MAAO,KoB8mOO,QpB9mOP,EoB8mOiB,CpB9mOjB,C;;QoBgnOd,OAAO,Q;O;KA  
pBX,C;mFAuBA,yB;MAAA,sE;MARjHI,8D;MpBtkHJ,iB;MoB2nOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAA  
e,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArkHG,cAAR,iBAAQ,C;QAqkHhB,aAAU,  
CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBroOG,MAAO,KoBqoOO,QpBroOP,EoBqoO  
iB,CpBroOjB,C;;QoBuoOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAPkHI,8D;MpB9kHJ,iB;MoBkpO  
A,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,  
OApHG,cAAR,iBAAQ,C;QAolHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB  
5pOG,MAAO,KoB4pOO,QpB5pOP,EoB4pOiB,CpB5pOjB,C;;QoB8pOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;M  
AAA,sE;MANlHI,8D;MpBtlHJ,iB;MoByqOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,e  
AAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnmHG,cAAR,iBAAQ,C;QAmHhB,aAAU,CAAV,iB;UACI,QA  
AQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnrOG,MAAO,KoBmrOO,QpBnrOP,EoBmrOiB,CpBnrOjB,C;;Qo

BqrOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAloHI,8D;MAkoHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhpHG,cAAR,iBAAQ,C;QAgpHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjpHI,8D;MAipHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/pHG,cAAR,iBAAQ,C;QA+pHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhqHI,8D;MAgqHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9qHG,cAAR,iBAAQ,C;QA8qHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/qHI,8D;MA+qHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7rHG,cAAR,iBAAQ,C;QA6rHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA9tHI,8D;MpBnjHJ,iB;MoBixOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5uHG,cAAR,iBAAQ,C;QA4uHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBzxOG,MAAO,KoByxOO,QpBzxOP,EoByxOiB,CpBzxOjB,C;;QoB2xOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3uHI,8D;MpB3jHJ,iB;MoBsyOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzvHG,cAAR,iBAAQ,C;QAyvHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9yOG,MAAO,KoB8yOO,QpB9yOP,EoB8yOiB,CpB9yOjB,C;;QoBgZOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAxvHI,8D;MpBnkHJ,iB;MoB2zOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7wHG,cAAR,iBAAQ,C;QAswHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn0OG,MAAO,KoBm0OO,QpBn0OP,EoBm0OiB,CpBn0OjB,C;;QoBq0Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MARwHI,8D;MpB3kHJ,iB;MoBg1OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAAnHG,cAAR,iBAAQ,C;QAmxHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx1OG,MAAO,KoBw1OO,QpBx1OP,EoBw1OiB,CpBx1OjB,C;;QoB01Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MALzHI,8D;MpB9jHJ,iB;MoBg3OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA0hHG,cAAR,iBAAQ,C;QAg0HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx3OG,MAAO,KoBw3OO,QpBx3OP,EoBw3OiB,CpBx3OjB,C;;QoB03Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA/zHI,8D;MpBtkHJ,iB;MoBq4OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA70HG,cAAR,iBAAQ,C;QA60HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB74OG,MAAO,KoB64OO,QpB74OP,EoB64OiB,CpB74OjB,C;;QoB+4Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA50HI,8D;MpB9kHJ,iB;MoB05OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA11HG,cAAR,iBAAQ,C;QA01HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBl6OG,MAAO,KoBk6OO,QpBl6OP,EoBk6OiB,CpBl6OjB,C;;QoB06Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAZ1HI,8D;MpBtlHJ,iB;MoB+6OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7vHG,cAAR,iBAAQ,C;QAU2HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBv7OG,MAAO,KoBu7OO,QpBv7OP,EoBu7OiB,CpBv7OjB,C;;QoBy7Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA74HI,8D;MA74HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA15HG,cAAR,iBAAQ,C;QAK5HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAN5HI,8D;MAM5HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5HG,cAAR,iBAAQ,C;QA+5HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAh6HI,8D;MAG6HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA56HG,cAAR,iBAAQ,C;QA46HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA76HI,8D;MA66HJ,sC;QAYiB,Q;QAFb,I

AAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz7HG,cAAR,iBAAQ,C;  
QAY7HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;  
YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MA19HI,8D;MA09HJ,kD;QAcIB,Q;Q  
AFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx+HG,cAAR,iB  
AAQ,C;QAw+HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,  
QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,y  
B;MAAA,sE;MAz+HI,8D;MAy+HJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,  
sBAAK,CAAL,CAAT,C;QACF,OAv/HG,cAAR,iBAAQ,C;QAU/HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBA  
AK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAA  
W,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAx/HI,8D;MAw/HJ,kD;QAcIB,Q;QAFb,IAAI,m  
BAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtgIG,cAAR,iBAAQ,C;QAsg  
IhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,  
CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;M  
AvgII,8D;MAugIJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,C  
AAT,C;QACF,OArhIG,cAAR,iBAAQ,C;QAqhIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,  
C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA  
AO,Q;O;KApBX,C;uGAuBA,yB;MATjII,8D;MASjIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB  
,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAkIG,cAAR,iBAAQ,C;QAKkIhB,aAAU,CAAV,iB;UACI,QAAQ,  
SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;Y  
ACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MAnkII,8D;MAmkIJ,kD;QAYiB,Q;QAFb,IAAI,m  
BAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/kIG,cAAR,iBAAQ,C;QA+kIh  
B,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CA  
AIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAhIII,8D;MAGII  
J,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5  
IIG,cAAR,iBAAQ,C;QA4IhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAA  
W,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;u  
GAqBA,yB;MA7III,8D;MA6IIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAA  
K,CAAL,CAAT,C;QACF,OAzmIG,cAAR,iBAAQ,C;QAymIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,C  
AAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAaIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;  
QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MACtB,UAAU,sBA  
AK,CAAL,C;MACG,OAIPIG,gBAAR,iBAAQ,C;MAkpIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QA  
CR,IPcP6P8D,YoCo6P1D,GpCp6P2E,KAAjB,EoCo6PpD,CpCp6PiF,KAA7B,CoCo6P1D,IAAJ,C;UAAa,MAAM,  
C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MACtB,UAAU,sBAAK,CA  
AL,C;MACG,OAzpIG,gBAAR,iBAAQ,C;MAypIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InB  
36P+D,amB26P3D,GnB36P6E,KAAIB,EmB26PrD,CnB36PmF,KAA9B,CmB26P3D,IAAJ,C;UAAa,MAAM,C;;M  
AEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;  
MACG,OAhqIG,gBAAR,iBAAQ,C;MAGqIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrCI9P4E,  
0BqCk9PxE,GrC7tP8B,KAAL,GAAiB,GArP8B,EqCk9PIE,CrC7tPwB,KAAL,GAAiB,GArP8B,CqCk9PxE,IAAJ,  
C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAae,OAAO,I;MACtB,U  
AAU,sBAAK,CAAL,C;MACG,OAvtIG,gBAAR,iBAAQ,C;MAuqIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,C  
AAL,C;QACR,InCz9P6E,0BmCy9PzE,GnC3uP8B,KAAL,GAAiB,KA9O+B,EmCy9PnE,CnC3uPwB,KAAL,GAA  
iB,KA9O+B,CmCy9PzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBA  
AJ,C;QAae,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAItIG,gBAAR,iBAAQ,C;MAktIhB,aAAU,C  
AAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAj  
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAae,MAAM,6B;MA  
CrB,UAAU,sBAAK,CAAL,C;MACG,OA7tIG,gBAAR,iBAAQ,C;MA6tIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA  
K,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE  
9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAae,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C

;MACG,OAxuIG,gBAAR,iBAAQ,C;MAwuIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAnvIG,gBAAR,iBAAQ,C;MAmvIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA1xIG,gBAAR,iBAAQ,C;MA0xIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAjyIG,gBAAR,iBAAQ,C;MAiyIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAxyIG,gBAAR,iBAAQ,C;MAwyIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA+yIG,gBAAR,iBAAQ,C;MA+yIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;qFAGX,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;mGAGJ,6B;MATgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;mGAGJ,6B;MATgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;qFAGJ,yB;MAAA,4F;MA9gJI,8D;MA8gJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAjJD,cAAR,iBAAQ,C;QAiiJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MA/hJI,8D;MA+hJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAljJD,cAAR,iBAAQ,C;QAKjJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAhJI,8D;MAgjJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAnkJD,cAAR,iBAAQ,C;QAmkjhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAjkJI,8D;MAikJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAplJD,cAAR,iBAAQ,C;QAolJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MAlnJI,8D;MAknJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OArjJD,cAAR,iBAAQ,C;QAqoJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MAnoJI,8D;MAmoJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAtpJD,cAAR,iBAAQ,C;QAspJhB,iBA

c,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MAppJI,8D;MAopJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAvqJD,cAAR,iBAAQ,C;QAuqJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MArqJI,8D;MAqqJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAxrJD,cAAR,iBAAQ,C;QAwrJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAttJI,8D;MAstJJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAxuJD,cAAR,iBAAQ,C;QAwuJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MATuJI,8D;MASuJJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAxvJD,cAAR,iBAAQ,C;QAwvJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MATvJI,8D;MASvJJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAxwJD,cAAR,iBAAQ,C;QAwvJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MATwJI,8D;MASwJJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAxzJD,cAAR,iBAAQ,C;QAwzJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MATxJI,8D;MASxJJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAY0JD,cAAR,iBAAQ,C;QAY0JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAV0JI,8D;MAU0JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA11JD,cAAR,iBAAQ,C;QA01JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAX1JI,8D;MAW1JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA32JD,cAAR,iBAAQ,C;QA22JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAZ2JI,8D;MAY2JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA53JD,cAAR,iBAAQ,C;QA43JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA15JI,8D;MA05JJ,uC;QAKB0B,UAEU,M;QAJhC,YA16JgB,cAAR,iBAAQ,C;QA26JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAL,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA36JI,8D;MA26JJ,uC;QAKB0B,UAEU,M;QAJhC,YA37JgB,cAAR,iBAAQ,C;QA47JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAL,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA57JI,8D;MA47JJ,uC;QAKB0B,UAEU,M;QAJhC,YA58JgB,cAAR,iBAAQ,C;QA68JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAL,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA78JI,8D;MA68JJ,uC;QAKB0B,UAEU,M;QAJhC,YA79JgB,cAAR,iBAAQ,C;QA89JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAL,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;6GAYBA,yB;MAAA,4F;MA9/JI,8D;MA8/JJ,uC;QAKB0B,Q;QAFtB,YA9gKgB,cAAR,iBAAQ,C;QA+gKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAhhKI,8D;MAghKJ,uC;QAKB0B,Q;QAFtB,YAhiKgB,cAAR,iBAAQ,C;QAiiKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MALiKI,8D;MAkiKJ,uC;QAKB0B,Q;QAFtB,YAljK

gB,cAAR,iBAAQ,C;QAmjKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,u  
BAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ  
,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAxBX,C;6GA0BA,yB;MAAA,4F;MAPjKI,8D;MAOj  
KJ,uC;QAKB0B,Q;QAFtB,YApkKgB,cAAR,iBAAQ,C;QAqkKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8  
B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA  
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAxBX,C;yHA0BA,  
yB;MAtmKI,8D;MAsmKJ,uC;QAIb0B,Q;QAFtB,YArnKgB,cAAR,iBAAQ,C;QAsnKhB,IAAI,QAAQ,CAAZ,C;U  
AAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA  
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAYBA,  
yB;MAvnKI,8D;MAunKJ,uC;QAIb0B,Q;QAFtB,YAtokgB,cAAR,iBAAQ,C;QAuoKhB,IAAI,QAAQ,CAAZ,C;U  
AAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA  
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAYBA,  
yB;MAxoKI,8D;MAwoKJ,uC;QAIb0B,Q;QAFtB,YAvpKgB,cAAR,iBAAQ,C;QAwPKhB,IAAI,QAAQ,CAAZ,C;  
UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,U  
AAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAYB  
A,yB;MAZpKI,8D;MAypKJ,uC;QAIb0B,Q;QAFtB,YAxqKgB,cAAR,iBAAQ,C;QAyqKhB,IAAI,QAAQ,CAAZ,C  
;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,  
UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;2GAY  
BA,yB;MA1sKI,8D;MA0sKJ,uC;QAKB0B,UAEU,M;QAJhC,YA1tKgB,cAAR,iBAAQ,C;QA2tKhB,IAAI,QAAQ,  
CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UAC  
I,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAYBA,  
yB;MA3tKI,8D;MA2tKJ,uC;QAKB0B,UAEU,M;QAJhC,YA3uKgB,cAAR,iBAAQ,C;QA4uKhB,IAAI,QAAQ,CA  
AZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,c  
AAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAYBA,y  
B;MA5uKI,8D;MA4uKJ,uC;QAKB0B,UAEU,M;QAJhC,YA5vKgB,cAAR,iBAAQ,C;QA6vKhB,IAAI,QAAQ,CA  
AZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,c  
AAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAYBA,y  
B;MA7vKI,8D;MA6vKJ,uC;QAKB0B,UAEU,M;QAJhC,YA7wKgB,cAAR,iBAAQ,C;QA8wKhB,IAAI,QAAQ,C  
AAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,  
cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,y  
B;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,  
kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBzjSO,W;QqB0jSP,kBAaKB,O;QACF,2B;QAAhB  
,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;  
QAEJ,OAAO,M;O;KAtBX,C;+FAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ  
,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBlISO,W  
;QqBmlSP,kBAaKB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,O  
AAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEJ,OAAO,M;O;KAtBX,C;+FAYBA,yB;MAAA,gD;MAAA,gE;MA  
AA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAA  
P,IAAb,C;QAA+B,8B;QAA5C,arB3mSO,W;QqB4mSP,kBAaKB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAg  
B,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEJ,OAAO,M;O;KAtB  
X,C;+FAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,  
OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBpoSO,W;QqBqoSP,kBAaKB,O;  
QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO  
,WAAI,WAAJ,C;;QAEJ,OAAO,M;O;KAtBX,C;6GAYBA,yB;MAAA,gD;MAAA,gE;MAI7KI,0D;MAK7KJ,gD;Q  
AmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C  
;QAA+B,8B;QAA5C,arB9pSO,W;QqB+pSP,kBAaKB,O;QACJ,OAr8KE,YAAR,iBAAQ,C;QAq8KF,mB;QAAA,  
kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAAL,CAA9B,C;UACd,  
MAAO,WAAI,WAAJ,C;;QAEJ,OAAO,M;O;KAxBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAP8KI,0D;MAo8

KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAA  
P,IAAb,C;QAA+B,8B;QAA5C,arBxrSO,W;QqByrSP,kBAAkB,O;QACJ,OA+v9KE,YAAR,iBAAQ,C;QAU9KF,mB  
;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;  
UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MA+9KI,0D;  
MA+9KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,  
CAAP,IAAb,C;QAA+B,8B;QAA5C,arBl+SO,W;QqBmtSP,kBAAkB,O;QACJ,OA+z+KE,YAAR,iBAAQ,C;QAY+K  
F,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA  
9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAx+  
KI,0D;MAw+KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,  
iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB5uSO,W;QqB6uSP,kBAAkB,O;QACJ,OA3/KE,YAAR,iBAAQ,C  
;QA2/KF,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KA  
AL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;mGA0BA,yB;MAAA,qD;MAAA,g  
E;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;Q  
ACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAl,aAAJ,C;QAA3C,arBtwSO,W;QqBuwSe,qB;QAAtB,iBAAC,C  
AAAd,wB;UACI,gBAAC,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,  
OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UA  
Ae,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAl,aA  
AJ,C;QAA5C,arB/xSO,W;QqBgySe,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UAAU,aAAV,EAAuB,sBAAK,  
KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,  
gE;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;  
QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAl,aAAJ,C;QAA5C,arBxzSO,W;QqByzSe,qB;QAAtB,iBAAC,C  
AAAd,wB;UACI,gBAAC,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,  
OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UA  
Ae,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACqC,kBAAXB,eAAkB,cAAIB,C;QAAgC,sBAAl,aA  
AJ,C;QAA7C,arBj1SO,W;QqBk1Se,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UAAU,aAAV,EAAuB,sBAAK,  
KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;iHAyBA,yB;MAAA,qD;MAAA,  
gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;  
QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAl,aAAJ,C;QAA3C,arB32SO,W;QqB42Se,qB;QAAtB,iBAAC,C  
AAAd,wB;UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,a  
AAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,  
mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA  
+B,sBAAl,aAAJ,C;QAA5C,arBr4SO,W;QqBs4Se,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UAAU,KAAV,EA  
AiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iH  
A0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBA  
AkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAl,aAAJ,C;QAA5C,arB/5SO,W;  
QqBg6Se,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CA  
A9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,  
uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACqC,k  
BAAXB,eAAkB,cAAIB,C;QAAgC,sBAAl,aAAJ,C;QAA7C,arBz7SO,W;QqB07Se,qB;QAAtB,iBAAC,CAAd,wB;  
UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;Q  
AEX,OAAO,M;O;KAvBX,C;iFA0BA,yB;MAxZA,gD;MAAA,gE;MAwZA,gD;QAgBW,sB;;UA+ZS,Q;UAHhB,IA  
AI,mBAAJ,C;YAAe,qBAAO,OAYZH,OAZZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UA  
A+B,sBAwZzB,OAxZyB,C;UAA5C,arBzjSO,W;UqB0jSP,kBAuZmB,O;UA+ZH,2B;UAAhB,OAAgB,cAAhB,C;Y  
AAgB,yB;YACZ,cAqZwB,SArZV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qB  
AAO,M;;;QAKZP,yB;O;KAhBJ,C;iFAMBA,yB;MAIZA,gD;MAAA,gE;MAkZA,gD;QAgBW,sB;;UAhZS,Q;UAH  
hB,IAAI,mBAAJ,C;YAAe,qBAAO,OAmZH,OAnZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,  
C;UAA+B,sBAkZzB,OAIzYB,C;UAA5C,arBlISO,W;UqBmlSP,kBAiZmB,O;UAhZH,2B;UAAhB,OAAgB,cAAh  
B,C;YAAgB,yB;YACZ,cA+YwB,SA/YV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UA





,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KafX,C;mFakBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KafX,C;mFakBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KafX,C;mFakBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KafX,C;mFakBA,yB;MpcTnTA,6B;MoCsnTA,sC;QAaoB,Q;QADhB,UpCxnTmC,coCwnTnB,CpCxnTmB,C;QoCynTnB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MpCt8TiD,coCs8TjD,GpCt8T2D,KAAK,GoCs8TzD,SAAS,OAAT,CpCt8ToE,KAAX,IAAf,C;;QoCw8TrD,OAAO,G;O;KAhBX,C;mFamBA,yB;MpczoTA,6B;MoCyoTA,sC;QAaoB,Q;QADhB,UpC3oTmC,coC2oTnB,CpC3oTmB,C;QoC4oTnB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MpCz9TiD,coCy9TjD,GpCz9T2D,KAAK,GoCy9TzD,SAAS,OAAT,CpCz9ToE,KAAX,IAAf,C;;QoC29TrD,OAAO,G;O;KAhBX,C;mFamBA,yB;Mpc5pTA,6B;MoC4pTA,sC;QAaoB,Q;QADhB,UpC9pTmC,coC8pTnB,CpC9pTmB,C;QoC+pTnB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC5+TiD,coC4+TjD,GpC5+T2D,KAAK,GoC4+TzD,SAAS,OAAT,CpC5+ToE,KAAX,IAAf,C;;QoC8+TrD,OAAO,G;O;KAhBX,C;mFamBA,yB;Mpc/qTA,6B;MoC+qTA,sC;QAaoB,Q;QADhB,UpCjrTmC,coCirTnB,CpCjrTmB,C;QoCkrTnB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC//TiD,coC+/TjD,GpC//T2D,KAAK,GoC+/TzD,SAAS,OAAT,CpC//ToE,KAAX,IAAf,C;;QoCigUrD,OAAO,G;O;KAhBX,C;mFamBA,yB;MnB/rTA,+B;MmB+rTA,sC;QAaoB,Q;QADhB,UnBhsTqC,eAAW,oBmBgsT/B,CnBhsT+B,CAAX,C;QmBisTrB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MnB/gUmD,emB+gUnD,GnB/gU8D,KAAK,KmB+gU5D,SAAS,OAAT,CnB/gUuE,KAAX,CAAhB,C;;QmBihUvD,OAAO,G;O;KAhBX,C;mFamBA,yB;MnBlfTA,+B;MmBktTA,sC;QAaoB,Q;QADhB,UnBntTqC,eAAW,oBmBmtT/B,CnBntT+B,CAAX,C;QmBotTrB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MnBliUmD,emBkiUnD,GnBliU8D,KAAK,KmBkiU5D,SAAS,OAAT,CnBliUuE,KAAX,CAAhB,C;;QmBoiUvD,OAAO,G;O;KAhBX,C;mFamBA,yB;MnBruTA,+B;MmBquTA,sC;QAaoB,Q;QADhB,UnBtuTqC,eAAW,oBmBsuT/B,CnBtuT+B,CAAX,C;QmBuuTrB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MnBrjUmD,emBqjUnD,GnBrjU8D,KAAK,KmBqjU5D,SAAS,OAAT,CnBrjUuE,KAAX,CAAhB,C;;QmBujUvD,OAAO,G;O;KAhBX,C;mFamBA,yB;MnBxvTA,+B;MmBwvTA,sC;QAaoB,Q;QADhB,UnBzvTqC,eAAW,oBmByvT/B,CnBzvT+B,CAAX,C;QmB0vTrB,2B;QAAGB,cAAhB,C;UAAgB,yB;UACZ,MnBxkUmD,emBwkUnD,GnBxkU8D,KAAK,KmBwkU5D,SAAS,OAAT,CnBxkUuE,KAAX,CAAhB,C;;QmB0kUvD,OAAO,G;O;KAhBX,C;IAmBA,kC;MA2DI,WpBv9TO,MAAO,KoBu9TG,cpBv9TH,EoBq6TH,KakDkB,OpBv9Tf,C;MoBw9Td,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CArDO,EAAnB,KAqDqB,CAAM,CAAN,CARDF,CAqDrB,C;;MArDT,OAuDO,I;K;IApDX,kC;MAkEI,WpB1+TO,MAAO,KoB0+TG,cpB1+TH,EoBi7TH,KAyDkB,OpB1+Tf,C;MoB2+Td,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAnB,KA4DqB,CAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpB7/TO,MAAO,KoB6/TG,cpB7/TH,EoB67TH,KAgEkB,OpB7/Tf,C;MoB8/Td,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAnB,KAmEqB,CAAM,CAAN,CAnEF,CAmErB,C;;MAnET,OAqEO,I;K;IAIEX,kC;MAgFI,WpBhhUO,MAAO,KoBghUG,cpBhhUH,EoBy8TH,KAuEkB,OpBhhUf,C;MoBihUd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAnB,KA0EqB,CAAM,CAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB19TA,iB;MoBk9TA,8C;QAWI,WpBv9TO,MAAO,KoBu9TG,cpBv9TH,EoBu9TS,KAAM,OpBv9Tf,C;QoBw9Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAANB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBr+TA,iB;MoBq+TA,8C;QAWI,WpB1+TO,MAAO,KoB0+TG,cpB1+TH,EoB0+TS,KAAM,OpB1+Tf,C;QoB2+Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV

,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAM,MAAM,CAAN,CAAnB,CAAJ,C;  
;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBx/TA,iB;MoBw/TA,8C;QAWI,WpB7/TO,MAAO,Ko  
B6/TG,cpB7/TH,EoB6/TS,KAAM,OpB7/Tf,C;QoB8/Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IA  
AlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAM,MAAM,CAAN,CAAnB,CAAJ,C;QAET,OA  
AO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3gUA,iB;MoB2gUA,8C;QAWI,WpBhhUO,MAAO,KoBghUG,c  
pBhhUH,EoBghUS,KAAM,OpBhhUf,C;QoBihUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,  
M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAM,MAAM,CAAN,CAAnB,CAAJ,C;QAET,OAAO,I;  
O;KAhBX,C;IAmBA,kC;MA8DoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iBpBplUJ,MAAO,KoBoIUsB,wBAn  
DzB,KAmDyB,EAAwB,EAAXB,CpBplUtB,EoBolUmD,SpBplUnD,CoBolUH,C;MACX,QAAQ,C;MACQ,OArD  
L,KAqDK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAG,SAAT,C;UAAoB,K;QACpB,IAAK,W  
AvDqB,GAuDP,uBAAK,UAAAL,EAAG,kBAAL,UAvDO,EAuDI,OAuDJ,CAuDrB,C;MAvDT,OAYDO,I;K;IAtd  
X,kC;MAuEoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iBpBzmUJ,MAAO,KoBymUsB,wBA5DzB,KA4DyB,E  
AAwB,EAAXB,CpBzmUtB,EoBymUmD,SpBzmUnD,CoBymUH,C;MACX,QAAQ,C;MACQ,OA9DL,KA8DK,W  
;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAG,SAAT,C;UAAoB,K;QACpB,IAAK,WAhEqB,GAGe  
P,uBAAK,UAAAL,EAAG,kBAAL,UAhEO,EAGeI,OAHEJ,CAgErB,C;MAhET,OAkEO,I;K;IA/DX,kC;MAGFoB,g  
B;MAHhB,gBAAGB,c;MACHB,WAAW,iBpB9nUJ,MAAO,KoB8nUsB,wBArEzB,KAqEyB,EAAwB,EAAXB,Cp  
B9nUtB,EoB8nUmD,SpB9nUnD,CoB8nUH,C;MACX,QAAQ,C;MACQ,OAveL,KAuEK,W;MAAhB,OAAGB,cA  
AhB,C;QAAGB,yB;QACZ,IAAI,KAAG,SAAT,C;UAAoB,K;QACpB,IAAK,WAZEqB,GAYEP,uBAAK,UAAAL,EA  
AK,kBAAL,UzEO,EAYEI,OAzEJ,CAyErB,C;MAzET,OA2EO,I;K;IAxEX,kC;MAyFoB,gB;MAHhB,gBAAGB,c  
;MACHB,WAAW,iBpBnpUJ,MAAO,KoBmpUsB,wBA9EzB,KA8EyB,EAAwB,EAAXB,CpBnpUtB,EoBmpUmD,  
SpBnpUnD,CoBmpUH,C;MACX,QAAQ,C;MACQ,OAHL,KAqFK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;Q  
ACZ,IAAI,KAAG,SAAT,C;UAAoB,K;QACpB,IAAK,WAlFqB,GAKFP,uBAAK,UAAAL,EAAG,kBAAL,UAlFO,E  
AkFI,OAlFJ,CAkFrB,C;MAIFT,OAoFO,I;K;+EAjFX,yB;MAAA,kF;MAAA,gE;MpB9kUA,iB;MoB8kUA,8C;QA  
coB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epBplUJ,MAAO,KoBoIUsB,wBAAN,KAAM,EAAwB,EAAX  
B,CpBplUtB,EoBolUmD,SpBplUnD,CoBolUH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAg  
B,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAAG,kBAAL,UA  
AV,EAAqB,OAARB,CAAJ,C;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpBnmUA,iB;M  
oBmmUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epBzmUJ,MAAO,KoBymUsB,wBAAN,KA  
AM,EAAwB,EAAXB,CpBzmUtB,EoBymUmD,SpBzmUnD,CoBymUH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OA  
AGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UA  
AL,EAAG,kBAAL,UAAV,EAAqB,OAARB,CAAJ,C;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,kF;MAA  
A,gE;MpBxnUA,iB;MoBwnUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epB9nUJ,MAAO,KoB  
8nUsB,wBAAN,KAAM,EAAwB,EAAXB,CpB9nUtB,EoB8nUmD,SpB9nUnD,CoB8nUH,C;QACX,QAAQ,C;QA  
CQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,U  
AAU,uBAAK,UAAAL,EAAG,kBAAL,UAAV,EAAqB,OAARB,CAAJ,C;QAET,OAAO,I;O;KAIBX,C;8EAqBA,yB  
;MAAA,kF;MAAA,gE;MpB7oUA,iB;MoB6oUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epBn  
pUJ,MAAO,KoBmpUsB,wBAAN,KAAM,EAAwB,EAAXB,CpBnpUtB,EoBmpUmD,SpBnpUnD,CoBmpUH,C;Q  
ACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAG,SAAT,C;YAAoB,K;UAC  
pB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAAG,kBAAL,UAAV,EAAqB,OAARB,CAAJ,C;QAET,OAAO,I;O;KAI  
BX,C;IAqBA,kC;MA2DI,WpBvtUO,MAAO,KoButUG,cpBvtUH,EoBqqUH,KAKdKB,KpBvtUf,C;MoBwtUd,WA  
AW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CAr  
DO,EAAAnB,KAqDqB,aAAM,CAAN,CArDF,CAqDrB,C;MArDT,OAuDO,I;K;IApDX,kC;MAKEI,WpB1uUO,M  
AAO,KoBoUUG,cpB1uUH,EoBirUH,KAYdKB,KpB1uUf,C;MoB2uUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CA  
AV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAAnB,KA4DqB,aAAM,CAAN,  
CA5DF,CA4DrB,C;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpB7vUO,MAAO,KoB6vUG,cpB7vUH,EoB6rUH,  
KAgEkB,KpB7vUf,C;MoB8vUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,  
WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAAnB,KAmEqB,aAAM,CAAN,CAnEF,CAMErB,C;MAnET,OAqE  
O,I;K;IALEX,kC;MAGFI,WpBhxUO,MAAO,KoBgxUG,cpBhxUH,EoBysUH,KAuEkB,KpBhxUf,C;MoBixUd,WA

AW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAnB,KA0EqB,aAAM,CAAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpBltUA,iB;MoBktUA,8C;QAWI,WpBvtUO,MAAO,KoButUG,cpBvtUH,EoButUS,KAAM,KpBvtUf,C;QoBwtUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBruUA,iB;MoBquUA,8C;QAWI,WpB1uUO,MAAO,KoB0uUG,cpB1uUH,EoB0uUS,KAAM,KpB1uUf,C;QoB2uUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBxvUA,iB;MoBwvUA,8C;QAWI,WpB7vUO,MAAO,KoB6vUG,cpB7vUH,EoB6vUS,KAAM,KpB7vUf,C;QoB8vUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3wUA,iB;MoB2wUA,8C;QAWI,WpBhxUO,MAAO,KoBgxUG,cpBhxUH,EoBgxUS,KAAM,KpBhxUf,C;QoBixUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAmBA,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpCr9UiD,SoCq9UjD,GpCr9U2D,KAAK,GoCq9UzD,OpCr9UoE,KAAX,IAAf,C;;MoCu9UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACjB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnBh+UmD,UmBg+UnD,GnBh+U8D,KAAK,KmBg+U5D,OnBh+UuE,KAAAX,CAAhB,C;;MmBk+UvD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpCj//UiD,SoCi//UjD,GpCj//U2D,KAAK,GAAW,CDqP5C,SqC4vUxB,OrC5vUkC,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;;MoCm//UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpC//UiD,SoC+//UjD,GpC//U2D,KAAK,GAAW,CCsP5C,SmCywUxB,OnCzwUkC,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;MoCigVrD,OAAO,G;K;+EAGX,yB;MAAA,0C;MpClsUA,6B;MoCksUA,4B;QAOL,OpC/rUmC,coC+rUpB,IAAR,iBAAQ,CpC/rUoB,C;O;KoCwrUvC,C;+EAUA,yB;MAAA,0C;MnB7rUA,+B;MmB6rUA,4B;QAOL,OnB1rUsC,emB0rUvB,IAAR,iBAAQ,CnB1rUuB,C;O;KmBmrU1C,C;+EAUA,yB;MAAA,sC;MpCttUA,6B;MoCstUA,iBAOiB,yB;QrCnzUb,6B;eqCmzUa,c;UAAE,OrC1yUoB,cqC0yUpB,ErC1yU8B,KAAL,GAAiB,GAAtB,C;S;OqC0yUtB,C;MAPjB,4B;QA7iBoB,Q;QADhB,UpC9pTmC,coC8pTnB,CpC9pTmB,C;QoC+pTnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC5+TiD,coC4+TjD,GpC5+T2D,KAAK,GAAW,CDqP5C,cqCuvTf,OrCvvTyB,KAAL,GAAiB,GAAtB,C;CrP4C,MAAX,IAAf,C;;QoC+hVrD,OAjjBO,G;O;KA0iBX,C;+EAUA,yB;MAAA,sC;MpChuUA,6B;MoCguUA,iBAOiB,yB;QnC5zUb,6B;emC4zUa,c;UAAE,OnCnzUoB,cmCmzUpB,EnCnzU8B,KAAL,GAAiB,KAAtB,C;S;OmCmzUtB,C;MAPjB,4B;QApiBoB,Q;QADhB,UpCjrTmC,coCirTnB,CpCjrTmB,C;QoCkrTnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC//TiD,coC+/TjD,GpC//T2D,KAAK,GAAW,CCsP5C,cmCywTf,OnCzwTyB,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;QoCyiVrD,OAXiBO,G;O;KAiiBX,C;IC/IVA,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,kC;MAQoB,UACL,M;MAHX,aAAa,eAAU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,oC;MAQoB,UACL,M;MAHX,aAAa,iBAAY,cAAZ,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MrCAiD,SqCAjD,GrCA2D,KAAK,GqCAzD,OrCAoE,KAAX,IAAf,C;;MqCERD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MpBXmD,UoBwnD,GpBX8D,KAAK,KoBW5D,OpBXuE,KAAX,CAAhB,C;;MoBavD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MrC5BiD,SqC4BjD,GrC5B2D,KAAK,GAAW,CDqP5C,SsCzNxB,OtCyNkC,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;;MqC8BrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MrC1CiD,SqC0CjD,GrC1C2D,KAAK,GAAW,CCsP5C,SoC5MxB,OpC4MkC,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;MqC4CrD,O

AAO,G;K;IC3GX,wB;MAMI,OtCuCkE,YsCvCvD,CtCuCwE,KAAjB,EsCvCID,CtCuC+E,KAA7B,CsCvCvD,KA  
AJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBsCmE,aqBtCxD,CrBsC0E,KAAIB,EqBtCnD,CrBsCiF,KA  
A9B,CqBtCxD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OvCKgF,0BuCLrE,CvC0P2B,KAAL,GAA  
iB,GArP8B,EuCLhE,CvC0PsB,KAAL,GAAiB,GArP8B,CuCLrE,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;  
MAMI,OrCIf,0BqCjTe,CrCkP2B,KAAL,GAAiB,KA9O+B,EqCjJe,CrCkPsB,KAAL,GAAiB,KA9O+B,CqCjTe,K  
AAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAA  
M,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,E  
AAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAA  
M,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,O  
AAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAA  
U,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,O  
AAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,  
SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,  
OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;M  
AOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,  
CAAX,C;;MACvB,OAAO,G;K;IAGX,wB;MAMI,OtCjFkE,YsCiFvD,CtCjFwE,KAAjB,EsCiFID,CtCjF+E,KAA7  
B,CsCiFvD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBIFmE,aqBkFxD,CrBIF0E,KAAIB,EqBkFn  
D,CrBIFiF,KAA9B,CqBkFxD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OvCnHgF,0BuCmHrE,CvC  
kl2B,KAAL,GAAiB,GArP8B,EuCmHhE,CvCkIsB,KAAL,GAAiB,GArP8B,CuCmHrE,KAAJ,GAAY,CAAZ,GAA  
mB,C;K;IAG9B,wB;MAMI,OrCpHiF,0BqCoHtE,CrC0H2B,KAAL,GAAiB,KA9O+B,EqCoHjE,CrC0HsB,KAAL,  
GAAiB,KA9O+B,CqCoHtE,KAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAA  
O,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;  
QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;  
MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;  
MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;I  
AUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GA  
AN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAA  
V,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MAD  
V,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;M  
ACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,  
MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IC7OX,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,  
2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2  
BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GA  
AyB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAYB,SAAK,M;K;IAGzC,4B;MASI,IAAI,m  
BAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mB  
AAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,kC;MAOI,OAAW,m  
BAAJ,GAAe,IAAf,GAAYB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAYB,SAAK,K;K;gFA  
GzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;gFAYA,yB;MAAA,mC;  
MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;IAYA,sC;;QASQ,OAAc,WAAP,MAAO,EAAS  
,SAAT,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QASQ,OAAc,YAAP,M  
AAO,EAAU,SAAV,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,  
mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAA  
A,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc  
,WAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,YAAP,MAA  
O,EAAU,SAAV,C;K;oFAGIB,8B;MASI,OAAO,WAAP,IAAX,IAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MASI,  
OAAO,WAAP,IAAX,IAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAMI,OAAO,2BxC8F4B,SwC9FnB,KxC8F6B,K  
AAL,GAAiB,GAAtB,CwC9F5B,C;K;IAGX,uC;MAMI,OAAO,2BxC+F8B,UAAW,oBwC/FhC,KxC+F2B,KAAK,  
CAAL,UAAW,CwC9F9B,C;K;IAGX,uC;MAMI,OAAO,2BvC0G8B,UAAW,oBuC1GhC,KvC0G2B,KAAK,CAAL,

iBAAN,CuC1G9B,C;K;IAGX,uC;MAMY,Q;MAAD,cAAC,OtBO4C,UsBP5C,KtB0kD,yBsBPxC,EtBOWC,CAA  
N,CsBP7C,wBAA8B,2BAA9B,Q;MAAA,W;QAAqC,oCvCsKR,SuCTKiB,KtB+FIB,KjBuEW,QAAV,CuCTKQ,C;;  
MAA5C,a;K;IAGJ,uC;MAMI,OAAO,2BtC2D4B,SsC3DnB,KtC2D6B,KAAL,GAAiB,KAAtB,CsC3D5B,C;K;IAG  
X,uC;MAMI,OAAO,2BtC4D8B,UAAW,oBsC5DhC,KtC4D2B,KAAK,CAAL,YAAN,CsC5D9B,C;K;IAGX,kC;M  
ASI,OAAO,uCAAgB,yBxCqCY,SwCrCI,SxCqCM,KAAL,GAAiB,GAAtB,CwCrCZ,ExCqCY,SwCrCmB,ExCqC  
T,KAAL,GAAiB,GAAtB,CwCrCZ,EAA4C,EAA5C,C;K;IAG3B,kC;MASI,OAAO,uCAAgB,yBAAGB,SAAhB,EA  
AsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,kC;MASI,OAAO,wCAAiB,yBAAGB,SAAhB,EAA5B,EAAtB,M;K;IAG  
5B,kC;MASI,OAAO,uCAAgB,yBtCEY,SsCFI,StCEM,KAAL,GAAiB,KAAtB,CsCFZ,EtCEY,SsCFmB,EtCET,K  
AAL,GAAiB,KAAtB,CsCFZ,EAA4C,EAA5C,C;K;IAG3B,gC;MAMI,OAAO,uCAAgB,yBAAGB,cAAhB,EAA5B,  
eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG3B,gC;MAMI,OAAO,wCAAiB,yBAAGB,cAAhB,EAA5B,eAAtB,  
EAA8B,cAAD,aAA7B,C;K;IAG5B,iC;MAMI,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAA  
gB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,GAAy,CAAhB,GAAmB,IAAnB,GAA6B,CAAC,IA  
AD,IAA1D,C;K;IAG3B,iC;MAMI,oBAAoB,kBAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,wCAAiB,yBAAG  
B,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,cAAy,CAAhB,GAAmB,IAAnB,GAA8B,IAAD,aAA1D,C;K;IA  
G5B,iC;MAQI,IxCvUgF,OBwCuU5E,ExClFkC,KAAL,GAAiB,GArP8B,EwCuUtE,6BAAM,UxCIFsB,KAAL,GA  
AiB,GArP8B,CwCuU5E,KAAJ,C;QAA2B,OAAO,iCAAU,M;MACHC,WxCjDuB,SwCiD5B,SxCjDsC,KAAL,GA  
AiB,GAAtB,C;MwCiDV,YAAK,W;MAA9B,OvCzI6D,oBAhJP,SAAU,CDwO7B,SwCiDV,ExCjDoB,KAAL,GA  
AiB,GAAtB,CCxO6B,MAAK,GDAK,KCAO,KAAZ,IAAf,CAGJO,C;K;IuC4IjE,iC;MAQI,IvCnUkE,YuCmU9D,E  
vCnU+E,KAAjB,EUcMuxD,4BAAK,UvCnUgF,KAA7B,CuCmU9D,KAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C  
,OvCrJ6D,cuCqJtD,SvCrJsD,EAhJP,SuCqStC,EvCrSgD,KAAK,GAAy,CuCqS5D,WvCrS4D,MAAZ,IAAf,CAGJO  
,C;K;IuCwJjE,iC;MAQI,ItBvUmE,asBuU/D,EtBvUiF,KAAIB,EsBuUzD,6BAAM,UtBvUiF,KAA9B,CsBuU/D,KA  
AJ,C;QAA2B,OAAO,kCAAW,M;MAC7C,OtBjK+D,iBsBiKxD,StBjKwD,EA7IP,UsB8SxC,EtB9SmD,KAAK,UA  
AY,CjBmQ/C,UAAW,oBAAL,CuC2CtB,WvC3CsB,MAAK,CAAL,iBAAN,CiBnQ+C,MAAZ,CAAhB,CA6IO,C;  
K;IsBoKnE,iC;MAQI,ItCnWiF,OBsCmW7E,EtCrHkC,KAAL,GAAiB,KA9O+B,EsCmWvE,8BAAO,UtCrHqB,KA  
AL,GAAiB,KA9O+B,CsCmW7E,KAAJ,C;QAA4B,OAAO,iCAAU,M;MACjC,WtCpFuB,SsCoF5B,StCpFsC,KA  
AL,GAAiB,KAAtB,C;MsCoFV,YAAK,W;MAA9B,OvC7K6D,oBAhJP,SAAU,CCyO7B,SsCoFV,EtCpFoB,KAA  
L,GAAiB,KAAtB,CDzO6B,MAAK,GCAK,KDAO,KAAZ,IAAf,CAGJO,C;K;IuCgLjE,kD;MAUI,OvCzWkE,YuC  
yWvD,SvCzWwE,KAAjB,EUcYWhD,YvCzW6E,KAA7B,CuCyWvD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,  
kD;MAUI,OtB9WmE,asB8WxD,StB9W0E,KAAIB,EsB8WjD,YtB9W+E,KAA9B,CsB8WxD,IAAJ,GAAyB,YAA  
zB,GAA2C,S;K;IAGtD,kD;MAUI,OxCnZgF,OBwCmZrE,SxC9J2B,KAAL,GAAiB,GArP8B,EwCmZ9D,YxC9JoB  
,KAAL,GAAiB,GArP8B,CwCmZrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OtCxZiF,OBsCwZtE,S  
tC1K2B,KAAL,GAAiB,KA9O+B,EsCwZ/D,YtC1KoB,KAAL,GAAiB,KA9O+B,CsCwZtE,IAAJ,GAAyB,YAAzB  
,GAA2C,S;K;IAGtD,iD;MAUI,OvC7ZkE,YuC6ZvD,SvC7ZwE,KAAjB,EUc6ZhD,YvC7Z6E,KAA7B,CuC6ZvD,I  
AAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtBlamE,asBkaxD,StBla0E,KAAIB,EsBkajD,YtBla+E,KAA  
9B,CsBkaxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OxCvcgF,OBwCucrE,SxCIN2B,KAAL,GAAi  
B,GArP8B,EwCuc9D,YxCINoB,KAAL,GAAiB,GArP8B,CwCucrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD  
;MAUI,OtC5ciF,OBsC4ctE,StC9N2B,KAAL,GAAiB,KA9O+B,EsC4c/D,YtC9NoB,KAAL,GAAiB,KA9O+B,CsC4  
ctE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,4D;MAUI,IvCjkdE,YuCid9D,YvCjd+E,KAAjB,EUcId/C,YvCjd4E  
,KAA7B,CuCid9D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAZB,C;MACvC,IvC  
ldkE,YuCkd9D,SvCld+E,KAAjB,EUcKdvD,YvCldoF,KAA7B,CuCkd9D,IAAJ,C;QAAyB,OAAO,Y;MACHC,IvCn  
dkE,YuCmD9D,SvCnd+E,KAAjB,EUcMdvD,YvCndoF,KAA7B,CuCmD9D,IAAJ,C;QAAyB,OAAO,Y;MACHC,O  
AAO,S;K;IAGX,4D;MAUI,ItBzdmE,asByd/D,YtBzdiF,KAAIB,EsBydhD,YtBzd8E,KAA9B,CsByd/D,IAAJ,C;QA  
AiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAZB,C;MACvC,ItB1dmE,asB0d/D,StB1diF,KAAI  
B,EsB0dxD,YtB1dsF,KAA9B,CsB0d/D,IAAJ,C;QAAyB,OAAO,Y;MACHC,ItB3dmE,asB2d/D,StB3diF,KAAIB,E  
sB2dxD,YtB3dsF,KAA9B,CsB2d/D,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,4D;MAUI,IxCjgBgF,0  
BwCigB5E,YxC5QkC,KAAL,GAAiB,GArP8B,EwCigB7D,YxC5QmB,KAAL,GAAiB,GArP8B,CwCigB5E,IAAJ,  
C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAZB,C;MACvC,IxClgBgF,OBwCkgB5E,SxC  
7QkC,KAAL,GAAiB,GArP8B,EwCkgBrE,YxC7Q2B,KAAL,GAAiB,GArP8B,CwCkgB5E,IAAJ,C;QAAyB,OAA

O, Y; MACHC, IxCngBgF, 0BwCmgB5E, SxC9QkC, KAAL, GAAiB, GARp8B, EwCmgBrE, YxC9Q2B, KAAL, GAAiB, GARp8B, CwCmgB5E, IAAJ, C; QAAyB, OAAO, Y; MACHC, OAAO, S; K; IAGX, 4D; MAUI, ItCzgBiF, 0BsCygB7E, YtC3RkC, KAAL, GAAiB, KA9O+B, EsCygB9D, YtC3RmB, KAAL, GAAiB, KA9O+B, CsCygB7E, IAAJ, C; QAAiC, MAAM, gCAAyB, oDAAiD, YAAjD, 8BAAoF, YAApF, MAAzB, C; MACvC, ItC1gBiF, 0BsC0gB7E, StC5RkC, KAAL, GAAiB, KA9O+B, EsC0gBtE, YtC5R2B, KAAL, GAAiB, KA9O+B, CsC0gB7E, IAAJ, C; QAAyB, OAAO, Y; MACHC, ItC3gBiF, 0BsC2gB7E, StC7RkC, KAAL, GAAiB, KA9O+B, EsC2gBtE, YtC7R2B, KAAL, GAAiB, KA9O+B, CsC2gB7E, IAAJ, C; QAAyB, OAAO, Y; MACHC, OAAO, S; K; IAGX, uC; MAcW, Q; MAJP, IAAI, 8CAAJ, C; QACI, OAAy, WAAL, SAAK, EAae, KAAf, C; MAEhB, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAyB, 4CAAyC, KAAzC, MAAzB, C; MAEvB, IvCthB8D, YuCshB9D, SvCthB+E, KAAjB, EuCshBvD, KAAM, MvCthB8E, KAA7B, CuCshB9D, K; QAA4B, OAAN, KAAM, M; QAC5B, IvCvhB8D, YuCuhB9D, SvCvhB+E, KAAjB, EuCuhBvD, KAAM, avCvhB8E, KAA7B, C uCuhB9D, K; UAAmC, OAAN, KAAM, a; UAC3B, gB; MAHZ, W; K; IAOJ, uC; MAcW, Q; MAJP, IAAI, 8CAAJ, C; QACI, OAAy, WAAL, SAAK, EAAGB, KAAhB, C; MAEhB, IAAI, KAAM, UAAV, C; QAAqB, MAAM, gCAAyB, 4CAAyC, KAAzC, MAAzB, C; MAEvB, ItBniB+D, asBmiB/D, StBniBiF, KAAiB, EsBmiBxD, KAAM, MtBniBgF, KAA9B, CsBmiB/D, K; QAA4B, OAAN, KAAM, M; QAC5B, ItBpiB+D, asBoiB/D, StBpiBiF, KAAiB, EsBoiBxD, KAAM, atBpiBgF, KAA9B, CsBoiB/D, K; UAAmC, OAAN, KAAM, a; UAC3B, gB; MAHZ, W; K; ICvIBJ, 2B; MAUoB, Q; MADhB, UAAGB, W; MACA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, MxCoDiD, SwCpDjD, GxCoD2D, KAAK, GwCpDzD, OxCoDoE, KAAX, IAAf, C; MwClDrD, OAAO, G; K; IAGX, 2B; MAUoB, Q; MADhB, UAAiB, 2B; MACD, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, MvBuCmD, UuBvCnD, GvBuC8D, KAAK, KuBvC5D, OvBuCuE, KAAX, CAAhB, C; MuBrCvD, OAAO, G; K; IAGX, 2B; MAUoB, Q; MADhB, UAAGB, W; MACA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, MxCoBiD, SwCpBjD, GxCoB2D, KAAK, GAAW, CDqP5C, SyCzQxB, OzCyQkC, KAAL, GAAiB, GAAtB, CCrP4C, MAAX, IAAf, C; MwClBrD, OAAO, G; K; IAGX, 2B; MAUoB, Q; MADhB, UAAGB, W; MACA, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, MxClId, SwCjJd, GxClI2D, KAAK, GAAW, CCsP5C, SuC1PxB, OvC0PkC, KAAL, GAAiB, KAAtB, CDtP4C, MAAX, IAAf, C; MwCFrD, OAAO, G; K; ICqCP, iD; MAAA, qE; MAAGB, 4B; MANpB, uC; MAMI, Y; K; IACA, 4D; MAAA, qE; MAAGC, wBAAM, OAAN, Q; MAPpC, uC; MAOI, Y; K; IACA, mE; MAAA, qE; MAAMd, 6BAAM, OAAN, EAae, KAAf, C; MARvD, uC; MAQI, Y; K; IACA, 0D; MAAA, qE; MAAiC, wBAAM, KAAN, Q; MATrC, uC; MASi, Y; K; ICtGJ, gC; K; ICuBoC, wC; 8BAAsC, O; K; CC0RtE, 6B; MASi, MAAM, yB; K; 0CAyDV, sB; MASi, OAAO, I; K; IC1Xf, gB; MAAA, oB; K; 8BAII, Y; MAA0B, oB; K; IAj9B, 4B; MAAA, 2B; QAAA, U; MAAA, oB; K; ICEA, yC; MAAA, e; MAAA, iB; MAAA, uB; K; IA AA, uC; MAAA, 0C; O; MAII, kE; MAEA, wF; MAEA, oF; MAEA, wE; MAEA, kE; MAEA, oF; MAEA, sF; MAEA, 8E; MA EA, wE; MAEA, sF; MAEA, uF; MAEA, iE; MAEA, 6E; MAEA, iE; MAEA, 2E; K; IA5BA, 8C; MAAA, 6B; MAAA, sC; K; IA EA, yD; MAAA, 6B; MAAA, iD; K; IA EA, uD; MAAA, 6B; MAAA, +C; K; IA EA, iD; MAAA, 6B; MAAA, yC; K; IA EA, 8C; MAAA, 6B; MAAA, sC; K; IA EA, uD; MAAA, 6B; MAAA, +C; K; IA EA, wD; MAAA, 6B; MAAA, gD; K; IA EA, o D; MAAA, 6B; MAAA, 4C; K; IA EA, iD; MAAA, 6B; MAAA, yC; K; IA EA, wD; MAAA, 6B; MAAA, gD; K; IA EA, wD; MAAA, 6B; MAAA, gD; K; IA EA, 6C; MAAA, 6B; MAAA, qC; K; IA EA, mD; MAAA, 6B; MAAA, 2C; K; IA EA, 6C; MA AA, 6B; MAAA, qC; K; IA EA, kD; MAAA, 6B; MAAA, 0C; K; IA hCj, mC; MAAA, +oB; K; IA AA, wC; MAAA, a; aAAA, O; UAAA, 2C; aAAA, kB; UAAA, sD; aAAA, gB; UAAA, oD; aAAA, U; UAAA, 8C; aAAA, O; UAAA, 2C; aAAA, gB; UAA A, oD; aAAA, iB; UAAA, qD; aAAA, a; UAAA, iD; aAAA, U; UAAA, 8C; aAAA, iB; UAAA, qD; aAAA, iB; UAAA, qD; aA AA, M; UAAA, 0C; aAAA, Y; UAAA, gD; aAAA, M; UAAA, 0C; aAAA, W; UAAA, +C; UAAA, uE; K; IAqCA, 4C; MA A, e; MAAA, iB; MAAA, uB; K; IA AA, 0C; MAAA, 6C; O; MAMI, 0E; MAEA, 0E; MAEA, 4E; K; IA JA, kD; MAAA, gC; M AAA, 0C; K; IA EA, kD; MAAA, gC; MAAA, 0C; K; IA EA, mD; MAAA, gC; MAAA, 2C; K; IA VJ, sC; MAAA, sI; K; IA A, 2C; MAAA, a; aAAA, Q; UAAA, +C; aAAA, Q; UAAA, +C; aAAA, S; UAAA, gD; UAAA, 0E; K; IA wB8B, gC; MAAC, oC; K; IAQE, 0B; MAAC, qB; QAAA, iD; MAAA, kB; K; IA EIc, sB; K; IA MA, 4B; K; IA XfQ, kD; MAAA, 8B; MA CI, aA AY, C; K; oDACZ, Y; MAAYB, oBAAQ, gBAAL, O; K; iDACrC, Y; MAAGD, Q; MAAI, IAAI, aAAQ, gBAAL, OA AhB, C; QAAA, OAAsB, iBAAL, iBAAJ, EAAL, yBAAJ, O; QAAkB, MAAM, 2BAAYB, UAAF, WAAvB, C; K; IA PhF, oC; MAEI, IAD8D, IAC9D, S; QACI, UAA0B, K; QAF0B, 2C; QAAA, QAAM, IAAN, C; eASxD, c; YATwD, OAStC, qBAaqB, KA ArB, C; eACIB, W; YAVwD, OAUzC, kBAAkB, KAAIB, C; eACf, Y; YAXwD, OAWxC, mBAAMb, KAAAnB, C; eAChB, W; YAZwD, OAYzC, kBAAkB, KAAIB, C; eACf, U; YAbwD, OAa1C, iBAAiB, KAAjB, C; eACd, W; YAdwD, OAzcC, k BA AkB, KAAIB, C; eACf, Y; YAfWd, OAexC, mBAAMb, KAAAnB, C; eAChB, a; YAhBwD, OAgBvC, oBAAoB, KAAp



7B,mBAAsC,I;MACtC,qBAAYC,I;MAEzC,yBAAGD,yBAAMb,Q;MAEnE,sBAAGD,I;K;wFAFhD,Y;MAAA,6B;K;0CAIA,Y;MAEY,kBADR,M;MAAA,U;MAAA,2C;QAAA,e;;QAES,gBADD,2CAAQ,yCAAR,gDAawD,IAAXD,6BAAiE,I;QACzD,sBpCwEd,S;QoC1EF,SpC2EG,S;;MoC3EH,a;K;iDAIJ,kB;MACI,kBAAC,IAAd,C;MACiC,oB;MCuBrB,Q;MADR,IdtBsB,MCsBtB,W;QADJ,mBACiB,I;;QADjB,mBAEY,QDvBc,MCuBd,+D;;MDvBZ,yC;MACA,2BAAMc,MAAO,kBAA1C,C;MAGA,OAAO,IAAP,C;QpCoCY,gBoCnCH,S;;QACD,iBAAiB,8B;QAGjB,IAAI,0BAAJ,C;UACI,qBAAC,e;;UAEd,oBAAQ,0B;UACR,wBAAy,kB;;UAIZ,cAAc,oB;UACd,IAAI,YAAY,yBAhB,C;YAAqC,M;UACrC,kBAAgB,O;UACbB,qBAAMb,I;;UAEnB,kBAAgB,I;UACbB,qBAAMb,S;;QAGvB,gC;QAEA,IAAI,wCAAJ,C;UAEL,YAAU,U;;UAGV,U;UAAA,0C;YETHB,8BDgDQ,WAAO,qBAAP,CChDR,C;YFSgB,a;;YAAA,a;UAAA,mB;YAEK,UEpBrB,oBDgDQ,WD5B+B,eC4B/B,CChDR,C;;UFqBgB,M;;K;mDAMhB,Y;MACI,kBAAkB,mB;MACIB,IAAI,uBAAuB,gBAAGB,IAA3C,C;QACI,uCAAQ,yCAAR,EAAmC,wCAA+B,WAA/B,C;;MAEvC,sBAAoB,mC;K;;IAM5B,iC;MAAA,qC;K;gGAEQ,Y;MvC0DyC,MAAM,6BuC1DjC,uCvC0D+D,WAA9B,C;K;yDuCxDnD,kB;MvCwD6C,MAAM,6BuCvDzC,uCvCuDuE,WAA9B,C;K;+CuCpDnD,Y;MAAkC,8C;K;;IARtC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IGyDA,mG;IAAA,yH;IAAA,6F;MAKW,kC;MAAS,4C;K;IALpB,sEAMQ,Y;MACI,Q;MAAA,sC;QAAiB,U;;MACjB,OAAO,oB;K;IARnB,6G;sJAJIA,iC;MAGBU,OAAK,SAAL,CAAiB,UAAjB,EAA6B,KAA7B,C;K;wJAEV,2C;MAiBU,OAAK,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,EAAuC,KAAvC,C;K;wJAEV,kD;MAKU,OAAK,SAAL,CAAiB,QAAjB,EAA2B,KAA3B,EAAkC,UAAIC,EAA8C,KAA9C,C;K;IAGc6C,oG;MAAA,mB;QAC3C,OAAK,iCAAL,CAAiB,kBAAjB,C;O;K;IA/BZ,6D;MA0BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAiB,UAAjB,EAA6B,IAA7B,C;;QA8D0B,Q;QAhE9B,4DAImD,0DAJnD,EAGe8B,qBA5DS,UA4DT,qCAhE9B,C;;K;IAwCmD,wH;MAAA,mB;QAC3C,OAAK,iCAAL,CAAiB,gBAAjB,EAA2B,kBAA3B,C;O;K;IAhCZ,yE;MA2BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,EAAuC,IAAvC,C;;QA0B0B,Q;QA5B9B,4DAImD,sEAJnD,EA4B8B,qBAxBs,UAWBT,qCA5B9B,C;;K;IASJ,gC;MAWK,kBAAD,M;MAAA,kBAAC,qEAAD,4DAA2C,S;K;6CAG/C,yB;MAAA,mG;MAAA,yH;MAAA,6F;QAKW,kC;QAAS,4C;O;MALpB,sEAMQ,Y;QACI,Q;QAAA,sC;UAAiB,U;;QACjB,OAAO,oB;O;MARnB,6G;MAAA,oC;QAKkC,Q;QAA9B,mEAA8B,oEAA9B,C;O;KALJ,C;IFC7HA,a;MAC6C,OAAA,MAAA,YAAW,CAAX,C;K;ICM3B,iC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;;;;;;;IAC/F,2B;MAAA,iD;MAAuB,oBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sC;MAAA,iD;MAAuC,oBAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,oC;MAAA,iD;MAAwC,oBAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAI+B,mC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;;;;;;;IACnG,+B;MAAA,mD;MAAuB,sBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,0C;MAAA,mD;MAAuC,sBAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,wC;MAAA,mD;MAAwC,sBAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAGsC,0C;MAA0D,qBAAU,OAAV,EAAmB,KAAhB,C;;K;;IACg,sC;MAAA,0D;MAAuB,6BAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,iD;MAAA,0D;MAAuC,6BAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,+C;MAAA,0D;MAAwC,6BAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAG8C,kD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACxG,8C;MAAA,kE;MAAuB,qCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,yD;MAAA,kE;MAAuC,qCAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,uD;MAAA,kE;MAAwC,qCAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAG2C,+C;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACrG,2C;MAAA,+D;MAAuB,kCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sD;MAAA,+D;MAAuC,kCAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,oD;MAAA,+D;MAAwC,kCAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAG+C,4C;8BAAwD,O;;K;;IACvG,+C;MAAA,mE;MAAuB,sCAAK,IAAL,C;MAAvB,Y;K;IAGqD,yD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC/G,qD;MAAA,yE;MAAuB,4CAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,gE;MAAA,yE;MAAuC,4CAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,8D;MAAA,yE;MAAwC,4CAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAGmD,uD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC7G,mD;MAAA,uE;MAAuB,0CAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,8D;MAAA,uE;MAAuC,0CAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,4D;MAAA,uE;MAAwC,0CAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAI2C,wC;sCAAGe,O;;K;;IAC3G,2C;MAAA,+D;MAAuB,kCAAK,IAAL,C;MAAvB,Y;K;IAI0C,uC;8BAAwD,O;;K;;IACIG,0C;MAAA,8D;MAAuB,iCAAK,IAAL,C;MAAvB,Y;K;IAGwC,qC;8BAAwD,O;;K;;IACg,wC;MAAA,4D;MAAuB,+BAAK,IAAL,C;MAAvB,Y;K;IAIJ,wC;MACmD,mBAAM,OAAN,EAAe,KAAf,C;;K;;IAC/C,oC;MAAA,wD;MAAuB,sBAAK,IAAL,Q;MAAvB,Y;K;IACA,+C;MAAA,wD;MAAgC,2BAAK,OAAL,EAAc,IAAd,C;MAAhC,Y;K;IACA,+C;MAA



A,wD;MAAiD,IAAY,I;MAAzB,2BAaA,SAAR,OAAQ,CAAb,EAAyB,sDAAzB,C;MAApC,Y;K;IAG4C,yC;8BA  
AwD,O;;K;;IACpG,4C;MAAA,gE;MAAuB,mCAAK,IAAL,C;MAAvB,Y;K;IAIyC,sC;8BAAwD,O;;K;;IACjG,yC;  
MAAA,6D;MAAuB,gCAAK,IAAL,C;MAAvB,Y;K;IAGkD,sD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;  
IAC5G,kD;MAAA,sE;MAAuB,yCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,6D;MAAA,sE;MAAuC,yCA  
AK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,2D;MAAA,sE;MAAwC,yCAAK,SAAL,EAAgB,KAAhB,C;MAAx  
C,Y;K;IAG0D,8D;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACpH,0D;MAAA,8E;MAAuB,iDAAK,IAA  
L,EAAW,IAAX,C;MAAvB,Y;K;IACA,qE;MAAA,8E;MAAuC,iDAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IAC  
A,mE;MAAA,8E;MAAwC,iDAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;6FCIGJ,yB;MAEI,OAAU,GAAG,CA  
AC,QAAD,C;K;mFAGjB,oB;MAEI,OAAG,GAAG,GAAS,G;K;6ETVb,a;MAK8C,cAAU,C;K;6ECHxD,Y;MAG+  
C,S;K;IA6B/C,2B;MAG4D,0BA Ae,WAAf,C;K;IAE5D,mC;MAIwF,0BA Ae,WAAf,C;K;IAExF,mC;MAKwE,0BA  
Ae,WAAf,C;K;IAGxE,4B;MAI8D,Q;MAH1D,aAAkB,GAAL,O;MACtB,aAAkB,GAAL,O;MACtB,YAAiB,C;MAC  
jB,OAAO,QAAQ,MAAR,IAAkB,QAAQ,MAAJ,C;QAAyC,IAAI,KAAJ,IAAa,IAAI,YAAJ,EAAI,oBAAJ,O;;MA  
CtD,OAAO,G;K;IAIX,wD;MAMuC,Q;MALnC,aAAa,MAAO,OAAM,CAAN,EAAS,OAAT,C;MA0BpB,IAzBc,M  
AyBL,OAAL,KAAkB,SAAtB,C;QAzBsB,MA0BIB,UAlBU,MA0BS,O;;MAzBvB,YAAiB,MAAO,O;MACxB,IA  
AI,UAAU,KAAAd,C;QACI,gBAAGB,O;QACHB,OAAO,QAAQ,OAaf,C;UAAwB,OAAO,YAAP,EAAO,oBAAP,U  
AAkB,Y;;MAE9C,OAAO,M;K;IAGX,gD;MAKOB,UAAmB,M;MAJnC,aAAa,KAAM,Q;MACnB,MAAO,OAAP,  
IAAiB,UAAW,K;MAC5B,IAbc,KAAaL,OAAL,KAAkB,SAAtB,C;QAbqB,MACjB,UAdU,KAcS,O;;MAbvB,YAAiB  
,KAAM,O;MACP,4B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAY,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;M  
AC9C,OAAO,M;K;IAGX,yD;MAEoB,UAAgB,M;MADhC,YAAy,U;MACI,4B;MAAhB,OAAGB,cAAhB,C;QAA  
gB,yB;QAAY,IAAI,cAAJ,EAAI,sBAAJ,YAAe,O;;MAC3C,OAAO,G;K;oFAGX,oB;MACI,IAAI,IAAK,OAAL,K  
AAkB,SAAtB,C;QACI,YAAc,IAAK,O;;K;0EAI3B,wB;MAA+D,OAAA,MAAa,QAAO,GAAP,EAAy,OAAZ,C;K  
;IS/F5E,mC;MAOI,kBAAkB,MAAa,eAAc,SAAd,C;MAC/B,iBAAiB,MAAa,eAAc,IAAd,C;MAC9B,OAAW,gBA  
Ae,UAAAnB,GAA+B,SAA/B,GAAYC,CAAC,S;K;0ECUrD,2B;MAKyE,OAAA,MAAa,gBA Ae,IAAf,C;K;4EAyBt  
F,2B;MAKsE,OAAA,MAAa,eAAc,IAAd,C;K;kEAGnF,qB;MACgD,OAAA,MAAa,KAAK,UAAAS,GAAT,EAAc,I  
AAd,C;K;wEACHC,qB;MAAQ,OAAK,SAAY,a;K;0EACxB,qB;MAAQ,OAAK,SAAY,c;K;IC3D5D,0D;MAGI,O  
AAO,I;K;ICHX,sC;MAMsD,OAAA,SAAY,UAAAS,WAAW,KAAX,CAAT,C;K;IhDKIE,uC;MhBynBW,Q;MAAA,  
IgbnnBgB,KhBnnBZ,IAAS,CAAT,IgbnnBY,KhBnnBE,IAAS,wBAA3B,C;QAAA,OAAc,UgbnnBtB,KhBnnB  
sB,C;;QgBnnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB0nBW,Q;MAAA,I  
gbpnBgB,KhBonBZ,IAAS,CAAT,IgbpnBY,KhBonBE,IAAS,0BAA3B,C;QAAA,OAAc,UgbpnBtB,KhBonBsB,  
C;;QgBpnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB2nBW,Q;MAAA,IgBr  
nBgB,KhBqnBZ,IAAS,CAAT,IgBrnBY,KhBqnBE,IAAS,0BAA3B,C;QAAA,OAAc,UgBrnBtB,KhBqnBsB,C;;Q  
gBrnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB4nBW,Q;MAAA,IgBt  
nBgB,KhBsnBZ,IAAS,CAAT,IgBtBY,KhBsnBE,IAAS,0BAA3B,C;QAAA,OAAc,UgBtntB,KhBsnBsB,C;;QgBt  
nb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB6nBW,Q;MAAA,IgBvnBgB,KhB  
unBZ,IAAS,CAAT,IgBvnBY,KhBunBE,IAAS,0BAA3B,C;QAAA,OAAc,UgBvnBtB,KhBunBsB,C;;QgBvnBb,  
MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB8nBW,Q;MAAA,IgBxnBgB,KhBw  
nBZ,IAAS,CAAT,IgBxnBY,KhBwnBE,IAAS,0BAA3B,C;QAAA,OAAc,UgBxnBtB,KhBwnBsB,C;;QgBxnBb,M  
AAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB+nBW,Q;MAAA,IgBznBgB,KhBznB  
Z,IAAS,CAAT,IgBznBY,KhBznBE,IAAS,0BAA3B,C;QAAA,OAAc,UgBznBtB,KhBznBsB,C;;QgBznBb,MAA  
M,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhBgoBW,Q;MAAA,IgB1nBgB,KhB0nBZ,I  
AAS,CAAT,IgB1nBY,KhB0nBE,IAAS,0BAA3B,C;QAAA,OAAc,UgB1nBtB,KhB0nBsB,C;;QgB1nBb,MAAM,  
8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,wC;MhBioBW,Q;MAAA,IgB3nBgB,KhB2nBZ,IAA  
S,CAAT,IgB3nBY,KhB2nBE,IAAS,0BAA3B,C;QAAA,OAAc,UgB3nBtB,KhB2nBsB,C;;QgB3nBb,MAAM,8B  
AA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,2B;MAII,OAAO,cAAa,SAAb,C;K;oFAGX,yB;MAAA,  
gD;MAAA,4B;QAKI,OAAc,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAuC,  
OAAhC,SAAgC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAqC,OAA9B,SAA8B,C;O;KALzC,  
C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAc,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MA  
AA,4B;QAKI,OAAuC,OAAhC,SAAgC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAwC,OAAj

C,SAAiC,C;O;KAL5C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAyC,OAAIC,SAAkC,C;O;KAL7C,C;IAY  
W,2C;MAAA,8B;MAAS,uB;K;4FACW,Y;MAAQ,OAAA,gBAAY,O;K;6CAC3C,Y;MAAkC,OAAA,gBhB8nP/B,  
YAAQ,C;K;oDgB7nPX,mB;MAAgD,OAAy,WAAZ,gBAAY,EAAS,OAAT,C;K;iDAC5D,iB;MACI,oCAAA,2BA  
AkB,KAAIB,EAAYB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,E;K;mDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,u  
FAAD,CAAJ,C;QAAGC,OAAO,E;MACvC,OAAmB,UAAZ,gBAAY,EAAQ,OAAR,C;K;uDAEvB,mB;MAES,Q;  
MAAL,IAAI,eAAC,uFAAD,CAAJ,C;QAAGC,OAAO,E;MACvC,OAAmB,cAAZ,gBAAY,EAAY,OOAZ,C;K;;IA  
pB/B,6B;MAII,0C;K;IAqBJ,+C;MAaI,OAAy,kBAAL,SAAK,EAakB,KAAIB,C;K;IAqBhB,0C;MASI,OAAy,oB  
AAL,SAAK,C;K;IAehB,0C;MAYI,OAAy,oBAAL,SAAK,C;K;IAkBhB,2C;MAWI,OAAy,cAAL,SAAK,EAAC,K  
AAd,C;K;IAGhB,2C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EA  
Ac,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SA  
K,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,  
SAAK,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAd,C;K;IAGhB,4C;MAWI,OAAy,c  
AAL,SAAK,EAAC,KAAd,C;K;IAwHhB,sC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,sC;MAOI,OAAy,gBAAL,  
SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,u  
C;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gB  
AAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IA  
oFhB,sC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,sC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAA  
y,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;  
K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,  
OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;K;wFAsGhB,yB;MAAA,8C;MAAA,kF;Q  
AmB0E,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SA  
V,EAAGB,WAAhB,EAAG6B,iBAA7B,EAAGD,UAAhD,EAAG4D,QAA5D,C;QACA,OAAO,W;O;KArBX,C;wFAw  
BA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,  
WAAgB,gB;QACjI,UAAU,SAAV,EAAG0C,WAA1C,EAAGiF,iBAAjF,EAAGoG,UAApG,EAAGH,QAAGH,C;QACA,  
OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,a  
AAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAAG2C,WAA3C,EAAGmF,iBAAGnF,EAAGS,UAA  
tG,EAAGH,QAAGH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBkE,iC;UAAA,oB  
AAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QAC/H,UAAU,SAAV,EAAYC,WAAzC,E  
AA+E,iBAA/E,EAAGK,UAAIG,EAAG8G,QAA9G,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MA  
AA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UA  
AU,SAAV,EAAG0C,WAA1C,EAAGiF,iBAAjF,EAAGoG,UAApG,EAAGH,QAAGH,C;QACA,OAAO,W;O;KArBX,C;  
wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;U  
AAA,WAAgB,gB;QACnI,UAAU,SAAV,EAAG2C,WAA3C,EAAGmF,iBAAGnF,EAAGS,UAAtG,EAAGH,QAAGH,C;  
QACA,OAAO,W;O;KArBX,C;uFAwBA,yB;MAAA,8C;MAAA,kF;QAmBwE,iC;UAAA,oBAAYB,C;QAAG,0B;  
UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACrI,UAAU,SAAV,EAAG4C,WAA5C,EAAGqF,iBAAGrF,EAAG  
G,UAAxG,EAAGoH,QAAGpH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmB0E,iC;U  
AAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAAG6C,W  
AA7C,EAAGuF,iBAAGvF,EAAG0G,UAA1G,EAAGS,QAAGtH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,  
8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QA  
CjI,UAAU,SAAV,EAAG0C,WAA1C,EAAGiF,iBAAjF,EAAGoG,UAApG,EAAGH,QAAGH,C;QACA,OAAO,W;O;KA  
rBX,C;oFAwBA,qB;MAOI,OAAy,SAAY,Q;K;oFAG5B,qB;MAOI,OAAy,SAAY,Q;K;oFAG5B,qB;MAOI,OAA  
y,SAAY,Q;K;qFAG5B,qB;MAOI,OAAy,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;M  
kCl7B9C,eAAiB,I;MICK7BjB,OkCj7BO,K;K;qFICo7BX,qB;MAOI,OAAy,SAAY,Q;K;qFAG5B,qB;MAOI,OAA  
y,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,c;MAAT,YAA8B,SAAY,Q;MkC/8BjD,eAAiB,I;MIC+8BjB,OkC98BO,  
K;K;IICi9BX,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MkCx9B9C,eAAiB,I;MICw9BjB,OkCv9BO,K;K;I  
IC09BX,uC;MD5oCI,IAAI,ECspCI,WAAW,CDtpCf,CAAJ,C;QACI,cCqpCoB,0C;QDppCpB,MAAM,gCAAyB,O  
AAQ,WAAjC,C;;MCqpCV,OAAO,SAAS,SAAT,EAAGe,cAAU,OAAV,CAAF,C;K;IAGX,uC;MD1pCI,IAAI,ECsq  
CI,WAAW,CDpqCf,CAAJ,C;QACI,cCmqCoB,0C;QDIqCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCmqCV,OA

AO,SAAS,SAAT,EAAe,eAAW,OAAx,CAAf,C;K;IAGX,uC;MDxqCI,IAAI,ECkrCI,WAAW,CDlrCf,CAAJ,C;QA  
CI,cCirCoB,0C;QDhrCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCirCV,OAAO,SAAS,SAAT,EAAe,eAAS,OAA  
T,CAAf,C;K;IAGX,uC;MDtrCI,IAAI,ECgsCI,WAAW,CDhsCf,CAAJ,C;QACI,cC+rCoB,0C;QD9rCpB,MAAM,g  
CAAyB,OAAQ,WAAjC,C;;MC+rCH,WAAS,W;MAAT,YAAsB,gBAAgB,SAAhB,EAAAsB,OAAAtB,K;MkChhC7B  
,eAAiB,I;MICghCjB,OkC/gCO,K;K;IICkhCX,uC;MDpsCI,IAAI,EC8sCI,WAAW,CD9sCf,CAAJ,C;QACI,cC6sCo  
B,0C;QD5sCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MC6sCV,OAAO,SAAS,SAAT,EAAe,iBAAW,OAAx,CAA  
f,C;K;IAGX,uC;MDltCI,IAAI,EC4tCI,WAAW,CD5tCf,CAAJ,C;QACI,cC2tCoB,0C;QD1tCpB,MAAM,gCAAyB,  
OAAQ,WAAjC,C;;MC2tCV,OAAO,SAAS,SAAT,EAAe,iBAAY,OAAZ,CAAf,C;K;IAGX,uC;MDhuCI,IAAI,EC0  
uCI,WAAW,CD1uCf,CAAJ,C;QACI,cCyuCoB,0C;QDxuCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCyuCH,W  
AAS,c;MAAT,YAAyB,gBAAgB,SAAhB,EAAAsB,OAAAtB,EAA+B,KAA/B,C;MkC1jChC,eAAiB,I;MIC0jCjB,OkC  
zjCO,K;K;IIC4jCX,uC;MD9uCI,IAAI,ECwvCI,WAAW,CDxvCf,CAAJ,C;QACI,cCuvCoB,0C;QDtvCpB,MAAM,  
gCAAyB,OAAQ,WAAjC,C;;MCuvCH,WAAS,W;MAAT,YAAsB,SAAS,SAAT,EAAe,iBAAU,OAAV,CAAf,C;M  
kCxcK7B,eAAiB,I;MICwkCjB,OkCvkCO,K;K;IIC0kCX,uC;MD5vCI,IAAI,ECuwCI,WAAW,CDvwCf,CAAJ,C;Q  
ACI,cCswCoB,0C;QDrwCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCswCV,OAAO,gBAAgB,SAAhB,EAAAsB,O  
AAAtB,EAA+B,IAA/B,C;K;IAGX,sD;MAWI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MAC  
b,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,  
EAAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB  
,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,s  
D;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SAAN,EA  
AiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MACN,WAAS  
,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkC9pC9C,eAAiB,I;MIC8pCjB,OkC7pCO,K;K;IIC  
gqCX,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MACb,OAAy,SAAY,OAAM,SA  
AN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MACb,  
OAAy,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,uD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,E  
AAsC,gBAAtC,C;MACN,WAAS,c;MAAT,YAA8B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkCxsCjD,eAAiB,I  
;MICwsCjB,OkCvsCO,K;K;IIC0sCX,uD;MAUI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;M  
ACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkCttC9C,eAAiB,I;MICstCjB,OkCrtC  
O,K;K;IICwtCX,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,SAAl  
B,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz3CD,ejD03CD,OiD13CC,EjD03CQ,SiD13CR,EjD03CmB,OiD13Cn  
B,C;K;IjD63ChB,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,SA  
AlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz4CD,ejD04CD,OiD14CC,EjD04CQ,SiD14CR,EjD04CmB,OiD14C  
nB,C;K;IjD64ChB,wD;MAWkD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjF,oCAAA,2BAAkB,SA  
AlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz5CD,ejD05CD,OiD15CC,EjD05CQ,SiD15CR,EjD05CmB,OiD15C  
nB,C;K;IjD65ChB,wD;MAW8C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC7E,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz6CD,ejD06CD,OiD16CC,EjD06CQ,SiD16CR,EjD06CmB,OiD16  
CnB,C;K;IjD66ChB,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz7CD,ejD07CD,OiD17CC,EjD07CQ,SiD17CR,EjD07CmB,OiD17  
CnB,C;K;IjD67ChB,wD;MAWkD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjF,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz8CD,ejD08CD,OiD18CC,EjD08CQ,SiD18CR,EjD08CmB,OiD18  
CnB,C;K;IjD68ChB,wD;MAWoD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACnF,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz9CD,ejD09CD,OiD19CC,EjD09CQ,SiD19CR,EjD09CmB,OiD19  
CnB,C;K;IjD69ChB,yD;MAWsD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACrF,oCAAA,2BAAkB,S  
AAIB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz+CD,ejD0+CD,OiD1+CC,EjD0+CQ,SiD1+CR,EjD0+CmB,OiD1  
+CnB,C;K;IjD6+ChB,yD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,  
SAAlB,EAA6B,OAA7B,EAAAsC,gBAAtC,C;MiDz/CD,ejD0/CD,oBiD1/CC,EjD0/CQ,SiD1/CR,EjD0/CmB,OiD1/  
CnB,C;K;iFjD6/ChB,8B;MAKI,OAAy,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;iFAG5B,yB;MAwIA,iD;MAxIA  
,qC;QAKI,OAwo,gCAxIK,eAAy,OAAZ,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwo  
O,gCAxIK,gBAAa,OAAb,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwo,gCAxIK,gBA

AW,OAAx,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,mBAAY,OAAZ,C  
AwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,kBAaA,OAAb,EAwIL,C;O;KA7  
IX,C;gFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,kBAAc,OAAd,EAwIL,C;O;KA7IX,C;iFAQA,y  
B;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCaxIK,sBAaE,OAaf,CAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;M  
AxIA,qC;QAKI,OAwIO,gCaxIK,mBAAY,OAAZ,CAwIL,C;O;KA7IX,C;IAQA,sC;MAKI,OAAO,oBAAoB,SAA  
pB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAaHb,IAAP,  
CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mB  
AAO,QAAS,KAaHb,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAA  
wB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAaHb,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IA  
GX,sC;MAII,OAAO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EA  
AO,mBAAO,QAAS,KAaHb,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,  
mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAaHb,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,  
C;K;IAGX,sC;MAII,OAAO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SA  
AK,EAAO,mBAAO,QAAS,KAaHb,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;iFAGX,+B;MA  
KI,OAAy,SAAY,QAAO,QAAP,C;K;iFAG5B,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,  
QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;K  
ALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,  
yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;  
MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QA  
KI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBA  
AqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EA  
A2B,QAA3B,C;O;KALX,C;8FAQA,8B;MAKI,OAAy,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;IAoBL,2B;MAA  
sB,OAAA,CAAE,iBAAU,CAAV,C;K;IAP/C,2B;MAOI,IAAI,mBAAO,CAAX,C;QkDvwDY,elDuwDO,WkDvwD  
P,C;K;IID4zDhB,2B;MAQI,IAAI,mBAAO,CAAX,C;QAAC,UAAU,SAAV,C;K;IAGIB,wC;MAQI,IAAI,mBAAO,  
CAAX,C;QAAC,cAAc,SAAd,EAAoB,UAApB,C;K;IAGIB,gD;MAewD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,  
UAAe,gB;MACvF,oCAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAC,SAAd,EAAoB,S  
AApB,EAA+B,OAA/B,EAAwC,cAAxC,C;K;IAGJ,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB  
;MAChE,oCAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EA  
AoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB  
;MACjE,oCAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EA  
oB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAagC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;  
MAC/D,oCAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EA  
oB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;  
MAChE,oCAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAC,SAAd,EAA8C,SAA9C,EA  
AyD,OAAzD,EAaKE,cAAIE,C;K;IAGJ,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,o  
CAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAp  
B,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAamC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACIE,o  
CAAA,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAp  
B,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAChE,oC  
AAa,2BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,  
C;MACvB,KAAT,QAAS,C;K;iFAGb,iC;MkD1+DgB,elDi/DD,UkDj/DC,C;K;iFIDo/DhB,iC;MkDp/DgB,elD2/DD  
,UkD3/DC,C;K;iFID8/DhB,iC;MkD9/DgB,elDqgED,UkDrgEC,C;K;iFIDwgEhB,iC;MkDxgEgB,elD+gED,UkD/gE  
C,C;K;iFIDkhEhB,iC;MkDlhEgB,elDyhED,UkDzhEC,C;K;iFID4hEhB,iC;MkD5hEgB,elDmiED,UkDniEC,C;K;iF  
IDsiEhB,iC;MkDtiEgB,elD6iED,UkD7iEC,C;K;IIDgjEhB,yC;MAMI,IAAI,mBAAO,CAAX,C;QAAC,gBAAC,SA  
Ad,EAAoB,UAApB,C;K;IAGIB,+D;MAA0E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACzG,oCAAA,2  
BAakB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAC,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EA  
AwC,UAAxC,C;K;IAGJ,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,M  
AAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,

EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII  
,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAOH,k  
D;MAAA,wB;QAAW,qCAAK,KAAL,E;O;K;IAJIC,oC;MAII,OAAO,iBAAM,gBAAN,EAAY,gCAAZ,C;K;ImDn  
pEX,oB;MAAA,wB;MAEI,6B;MACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MA  
Cf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAA  
W,oBAAS,CAAT,CrC4BuB,IqC5BIC,IAA+B,C;;MAInC,qBAAqB,48C;MACrB,WAAW,mBAAmB,cAAnB,EAA  
mC,UAAAnC,EAA+C,IAA/C,C;MACX,YAAY,eAAS,IAAK,OAAL,GAAY,CAAZ,IAAT,C;MACZ,0BAAU,IAAV,  
e;QACI,MAAM,MAAI,CAAJ,IAAN,IAAe,MAAM,GAAN,IAAW,KAAK,GAAL,CAAX,I;;MAEnB,yBAoB,K;  
MAGpB,oBAAoB,m/D;MACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAIC,EAA8C,IAA9C,C;K;;IAvB/B,gC;  
MAAA,+B;QAAA,c;;MAAA,wB;K;IA2BA,qC;MAKkB,IAJP,I;MACH,WAAO,EAAP,C;QAAe,W;WACf,WAAO  
,IAAP,C;QAAgB,OAAI,CAAC,KAAO,CAAR,MAAc,CAAIB,GAAqB,QAAS,CAA9B,GAAqC,OAAS,E;;QAE1D  
,QAAM,KAAK,CAAL,IAAN,C;eACI,C;YAAK,eAAS,E;YAAAd,K;eACA,C;YAAK,OAAC,QAAS,CAAV,GAaiB,  
E;YAAtB,K;;YACQ,cAAS,E;YAHrB,K;;MAJR,W;K;IAYJ,qC;MAIL,SAAS,SrCPiC,I;MqCS1C,YAAY,kBAakB,  
sBAAS,kBAA3B,EAA8C,EAA9C,C;MACZ,YAAY,sBAAS,kBAAT,CAA2B,KAA3B,C;MACZ,WAAW,sBAAS,  
qBAAT,CAA8B,KAA9B,C;MACX,YAAY,kBAakB,IAAIB,EAAwB,KAAK,KAAL,IAAxB,C;MAEZ,OAAW,U  
AAS,EAAb,GAAYC,mDAAzC,GAAoD,K;K;IAG/D,8D;MAKiB,UAIE,M;MARf,aAAa,eAAS,YAAT,C;MACb,Y  
AAY,C;MACZ,UAAU,C;MACV,YAAY,C;MACC,yB;MAAb,OAAa,cAAb,C;QAAa,iC;QACT,aAAa,WAAW,IrC  
vBc,IqCuBzB,C;QACb,MAAM,MAAQ,CAAC,SAAW,EAAZ,KAAsB,K;QACpC,IAAI,SAAS,EAAb,C;UACI,OA  
AO,cAAP,EAAO,sBAAP,YAAkB,G;UACIB,MAAM,C;UACN,QAAQ,C;;UAER,gBAAS,CAAT,I;;MAGR,OAA  
O,M;K;IClEX,+B;MAII,eAAe,CAAC,iBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,  
M;MACIB,mBAAmB,2B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,sBAAK,KAAL,C;QACV,sBA  
AK,KAAL,EAAc,sBAAK,YAAL,CAAd,C;QACA,sBAAK,YAAL,EAAqB,GAArB,C;QACA,mC;;K;IjDbR,wB;M  
AOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAkBA,iB;MAIBA,uB;QAMI,OA  
kBO,MAAO,KAlBC,CAkBD,EAlBY,CAkBZ,C;O;KAXBIB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,M  
AAO,KATC,CASD,EATY,CASZ,C;O;KAlfB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,  
CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C  
;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA  
,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OA  
AO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OA  
AO,MAAO,KAAI,CAAN,EAaiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;Q  
AMI,OAAO,MAAO,KAAI,CAAN,EAaiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAA  
A,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KANIB,C;mFASA,mB;MAMW,UAA  
e,CAPeX,iBAoEc,CAPeD,MAAJ,GAoEe,CAPeF,GAoEkB,C;MAAZB,OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAo  
EM,CAPeN,GAAMB,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP  
,EAAU,CAAV,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CA  
AP,EAAU,CAAV,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAA  
A,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,  
UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAXHf,EAwH  
oB,CAXHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,Q  
AAA,KAAV,M;QAAiB,MAIV,MAAO,KAKIe,GAlIf,EAKIoB,CAlIpB,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,  
Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,  
EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAA  
U,QAAA,KAAV,M;QAAuB,UAAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GAAZ,GA6I2B,C;;MA  
CIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QA  
AiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C  
;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;  
MAGJd,OAAO,G;K;IAGX,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;mFAG9B,yB;  
MAkBA,iB;MAIBA,uB;QAMI,OAkBO,MAAO,KAlBC,CAkBD,EAlBY,CAkBZ,C;O;KAXBIB,C;mFASA,yB;MA

SA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O;KafIB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPed,MAAJ,GAoEe,CAPef,GAoEkB,C;MAAZB,OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAMB,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EAAW,CAAX,C;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAxHf,EAwHoB,CxHpB,C;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAAO,KAKIe,GAlIf,EAKIoB,CAlIpB,C;MAmId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GAAZ,GA6I2B,C;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;MAGJd,OAAO,G;K;IkDvaX,iB;MAAA,qB;MAEI,0BAA0B,gBACtB,EADsB,EACd,IADc,EACN,IADM,EACE,IADF,EACU,IADV,EACkB,IADIB,EAC0B,IAD1B,EACkC,IADIC,EAC0C,IAD1C,EACkD,IADID,EAC0D,IAD1D,EACkE,IADIE,EAC0E,IAD1E,EACkF,IADIF,EAC0F,IAD1F,EACkG,IADIG,EAC0G,IAD1G,EACkH,IADIH,EAC0H,IAD1H,EACkI,IADII,EAEtB,IAFsB,EAEd,IAFc,EAEN,IAFM,EAEe,IAFF,EAEU,IAFV,EAEkB,IAFIB,EAE0B,IAF1B,EAEkC,IAFIC,EAE0C,IAF1C,EAEkD,KAFID,EAE0D,KAF1D,EAEkE,KAFIE,EAE0E,KAF1E,EAEkF,KAFIF,EAE0F,KAF1F,EAEkG,KAFIG,EAE0G,KAF1G,E;K;IAF9B,6B;MAAA,4B;QAAA,W;MAAA,qB;K;IAQA,0C;MAKI,aAAa,C;MACb,UAAU,KAAM,OAAN,GAAa,CAAa,I;MACV,aAAa,E;MACb,YAAY,C;MACZ,OAAO,UAAU,GAajB,C;QACI,SAAS,CAAC,SAAS,GAAT,IAAD,IAAiB,CAAjB,I;QACT,QAAQ,MAAM,MAAN,C;QACR,IAAI,SAAS,KAAb,C;UACI,SAAS,SAAS,CAAT,I;aACR,IAAI,WA AU,KAAd,C;UACD,OAAO,M;UAEP,MAAM,SAAS,CAAT,I;MAEd,OAAO,UAAc,SAAS,KAAb,GAAoB,CAApB,GAA2B,CAArC,K;K;IAGX,mC;MAKI,SAAS,SvCEiC,I;MuCD1C,YAAY,kBAAkB,mBAAM,mBAAXB,EAAoC,EAApC,C;MACZ,WAAW,KAAK,mBAAM,mBAAN,CAAiB,KAajB,CAAL,I;MACX,OAAW,OAAO,EAAX,GAAe,IAAf,GAAYB,E;K;IAGpC,gC;MAII,OAAO,6BAAoB,C;K;IC7C/B,kB;MAAA,sB;MAEI,6B;MACA,8B;MACA,gC;MAKuB,UAAU,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAAiB,eAAS,GAAT,C;MAC E,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CxC2BuB,IwC3BIC,IAA+B,C;MAInC,qBAAqB,sW;MACrB,WAAW,mBAAmB,cAAnB,EAAmC,UAAnc,EAA+C,GAA/C,C;MACX,YAAY,eAAS,IAAK,OAAd,C;MACZ,0BAAU,IAAV,e;QACI,IAAI,QAAC,CAAT,C;UAAy,MAAM,GAAN,IAAW,KAAK,GAAL,C;UACIB,MAAM,GAAN,IAAW,MAAM,MAAI,CAAJ,IAAN,IAAe,KAAK,GAAL,CAAf,I;MAEpB,yBAAoB,K;MAGpB,kBAAkB,0U;MACIB,0BAAqB,mBAAmB,WAAnc,EAAgC,UAAhC,EAA4C,GA A5C,C;MAGrB,oBAAoB,i8B;MACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAIC,EAA8C,GAA9C,C;K;IA7B /B,8B;MAAA,6B;QAAA,Y;MAAA,sB;K;IAiCa,iC;MAII,OAAO,6BAAmB,C;K;IAG9B,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wBxCPM,awCON,C;MAApC,W;K;IAGJ,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wBx CdM,awCcN,C;MAApC,W;K;IAGJ,kC;MAQI,SAAS,SxCzBiC,I;MwC0B1C,YAAY,kBAAkB,oBAAO,kBAAZB, EAA4C,EAA5C,C;MAEZ,iBAAiB,oBAAO,kBAAP,CAAYB,KAAZB,C;MACjB,eAAe,aAAa,oBAAO,mBAAP,C AA0B,KAA1B,CAAa,GAAGD,CAAhD,I;MACf,WAAW,oBAAO,qBAAP,CAA4B,KAA5B,C;MAEX,IAAI,KAA K,QAAT,C;QACI,OAAO,C;MAGX,kBAAkB,OAAS,C;MAE3B,IAAI,gBAAe,CAAnB,C;QACI,YAAY,C;QACZ,

gBAAgB,U;QACbB,aAAU,CAAV,OAAa,CAAb,M;UACI,yBAAc,QAAS,KAAV,GAAqB,GAAIC,K;UACA,IAAI, YAAAY,EAAbB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;UACA,yBAAc,QAAS,KAAV,GAAqB,GAAIC,K;UA CA,IAAI,YAAAY,EAAbB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;;QAEJ,OAAO,C;;MAGX,IAAI,QAAQ,CA AZ,C;QACI,OAAO,W;;MAGX,eAAgB,KAAK,UAAI,I;MACHb,cAAgB,QAAQ,EAAZ,GAAbB,WAAW,CAAX, IAAIB,GAAoC,Q;MACHd,OAAQ,SAAU,IAAI,OAAJ,IAAV,CAAD,GAA2B,C;K;ICnGtC,0B;MAAA,8B;MACI, +BAA+B,gBAC3B,GAD2B,EACnB,GADmB,EACX,GADW,EACH,GADG,EACK,GADL,EACa,GADb,EACqB, GADrB,EAC6B,IAD7B,EACqC,IADrC,EAC6C,IAD7C,EACqD,IADrD,EAC6D,IAD7D,EACqE,IADrE,EAC6E,I AD7E,EACqF,IADrF,EAC6F,KAD7F,EACqG,KADrG,EAC6G,KAD7G,EACqH,KADrH,EAC6H,KAD7H,E;MA G/B,gCAAgC,gBAC5B,CAD4B,EACzB,CADyB,EACtB,CADsB,EACnB,CADmB,EACHB,CADgB,EACb,CADa, EACV,CADU,EACP,EADO,EACH,CADG,EACA,EADA,EACI,CADJ,EACO,CADP,EACU,EADV,EACc,EADD, EACKb,EADiB,EACsB,CADtB,EACyB,CADzB,EAC4B,CAD5B,EAC+B,CAD/B,EACKC,CADIC,E;K;;;IAJpC,s C;MAAA,qC;QAAA,oB;;MAAA,8B;K;IASA,qC;MACI,YAAAY,kBAAbB,4BAAE,wBAAjC,EAakD,SAaID,C;M ACZ,OAAO,SAAS,CAAT,IAAc,aAAO,4BAAE,wBAAf,CAA+B,KAA/B,IAAwC,4BAAE,yBAAf,CAAgC,KAAh C,CAAxC,IAAP,C;K;ICXzB,qC;MACI,OAAe,IAAR,8BAAGB,IAAhB,KACY,IAAR,8BAAGB,IADpB,C;K;ICCX, wC;MxCiBW,Q;MAAA,IwCXgB,KxCWZ,IAAS,CAAT,IwCXY,KxCWE,IAAS,2BAA3B,C;QAAA,OAAc,qBw CXtB,KxCWsb,C;;QwCXb,MAAM,8BAA0B,mCAAyB,gBAAzB,MAA1B,C;;MAAtC,W;K;ICRj,sC;MAEI,WA AW,S5CmC+B,I;M4CjC1C,IAAY,GAAR,oBAAGB,GAAbB,KAAkC,GAAR,oBAAGB,GAA1C,CAAJ,C;QACI,O AA8B,OAAtB,KAAK,CAAC,OAAO,CAAP,IAAD,IAAa,CAAb,IAAL,KAAsB,C;;MAGIC,IAAY,IAAR,oBAAGB ,IAAhB,KAAkC,IAAR,oBAAGB,IAA1C,CAAJ,C;QACI,OAAO,S;;MAEX,OAAO,wB;K;ICPX,wC;MpCqTe,WoC 7SY,KpC6SZ,IAAS,C;MAAT,S;QAAC,OoC7SF,KpC6SE,IAyGHT,gBAAR,iBAAQ,C;;MAzgHT,U;MAAA,S;QA AA,SAAsC,sBoC7StB,KpC6SsB,C;;QoC7Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,w C;MpCsTe,WoC9SY,KpC8SZ,IAAS,C;MAAT,S;QAAC,OoC9SF,KpC8SE,IAqGHT,gBAAR,iBAAQ,C;;MArgHT, U;MAAA,S;QAAA,SAAsC,sBoC9StB,KpC8SsB,C;;QoC9Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MA AtC,a;K;IAGJ,wC;MpCuTe,WoC/SY,KpC+SZ,IAAS,C;MAAT,S;QAAC,OoC/SF,KpC+SE,IAiGHT,gBAAR,iBAA Q,C;;MAjgHT,U;MAAA,S;QAAA,SAAsC,sBoC/StB,KpC+SsB,C;;QoC/Sb,MAAM,8BAA0B,iCAAuB,cAAvB,M AA1B,C;;MAAtC,a;K;IAGJ,wC;MpCwTe,WoChTY,KpCgTZ,IAAS,C;MAAT,S;QAAC,OoChTF,KpCgTE,IA6/G T,gBAAR,iBAAQ,C;;MA7/GT,U;MAAA,S;QAAA,SAAsC,sBoChTtB,KpCgTsB,C;;QoChTb,MAAM,8BAA0B,iC AAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IASO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K; K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAGD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC5 D,iB;MACI,oCAAa,2BAAbB,KAAIB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES, Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAGC,OAAO,E;MACvC,OpC0rBO,UoC1rBA,gBpC0rBR,QAAQ,EoC 1rBoB,OxE0OF,KoCgdIB,C;K;yDoCxrBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAGC,OAAO,E; MACvC,OpC66BO,coC76BA,gBpC66BR,QAAQ,EoC76BwB,OxEqON,KoCwsBIB,C;K;;IoCn8BnB,6B;MAMI,4 C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gB AAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAa,2BAAbB,KAA IB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QA AJ,C;QAAiC,OAAO,E;MACxC,OpCyqBO,UoCzqBA,gBpCyqBR,QAAQ,EoCzqBoB,OvD0NA,KmB+cpB,C;K;y DoCvqBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpC45BO,coC55BA,g BpC45BR,QAAQ,EoC55BwB,OvDqNJ,KmBusBpB,C;K;;IoCl7BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MA AS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAi D,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAa,2BAAbB,KAAIB,EAAyB,SAAzB,C;MACb,O AAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC ,OpCwpBO,UoCxpBA,gBpCwpBR,QAAQ,EoCxpBoB,OzE4IA,KqC4gBpB,C;K;yDoCtpBX,mB;MAES,Q;MAAL ,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpC24BO,coC34BA,gBpC24BR,QAAQ,EoC34BwB,O zEuIJ,KqCowBpB,C;K;;IoCj6BnB,8B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAA A,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAkD,OAAA,gBAAY,gBAAS,OAAT, C;K;mDAC9D,iB;MACI,oCAAa,2BAAbB,KAAIB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAE X,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAC,OAAO,E;MACzC,OpCuoBO,UoCvoBA,gBpCuoB





BAAJ,WAAc,CAAd,C;QACA,UAAU,I;;MAEd,OAAO,O;K;0CAGX,Y;MACI,qB;MACA,yBAAY,CAAZ,EAAe, SAAf,C;K;IAKiB,gE;MAAA,qB;QAAE,OAAM,gBAAN,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU ,8CAAV,C;K;IAKU,gE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAA U,8CAAV,C;K;6CAIX,Y;MAAqD,iD;K;mDAErD,mB;MAAoD,0BAAQ,OAAR,KAAoB,C;K;kDAExE,mB;MAC qB,Q;MAAA,6B;MAAjB,iBAAc,CAAd,yB;QACI,IAAI,wBAAI,KAAJ,GAAc,OAAd,CAAJ,C;UACI,OAAO,K;;; MAGf,OAAO,E;K;sDAGX,mB;MACI,iBAAc,sBAAd,WAA+B,CAA/B,U;QACI,IAAI,wBAAI,KAAJ,GAAc,OA Ad,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;iDAGX,Y;MAA6D,iCAAA,CAAb,C;K;yDAC7D,iB;MAAuE,sD AAIb,KAAjB,C;K;oDAGvE,8B;MAA4E,uCAAQ,IAAR,EAAC,SAAd,EAAYB,OAAzB,C;K;wDAE5E,8B;MAII,e AAe,0BAAa,SAAb,C;MACf,YAAO,UAAU,SAAV,I;M/DuDX,iBAAc,CAAd,UAAaB,KAAtB,U;Q+DtDiB,e;QAC A,iB;;K;2CAIjB,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;M AE9B,OAAO,oCAAA,uBAAc,IAAd,EAAoB,KAApB,C;K;6CAGxB,Y;MAG+B,OAAA,oCAAA,yBAAGB,IAAhB, C;K;IAG5C,kD;MAAA,oB;MACI,eACsB,C;MAcTb,cAIqB,E;K;yDAErB,Y;MAAkC,sBAAQ,gB;K;sDAE1C,Y; MAEW,Q;MADP,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MAcTb,eAAO,mBAAP,EAAO,2BAAP,O;MACA,O AAO,wBAAI,WAAJ,C;K;wDAGX,Y;MIE5CJ,IAAI,EkE6CU,gBAAQ,EIE7CIB,CAAJ,C;QACI,ckE4CwB,sE;QIE 3CxB,MAAM,6BAAsB,OAAQ,WAA9B,C;;MkE6CF,6BAAS,WAAT,C;MACA,eAAQ,W;MACR,cAAO,E;K;;IA OqB,6D;MAHpC,oB;MAGmD,wD;MAG3C,oCAAA,4BAAmB,KAAAnB,EAA0B,WAAyB,KAAAnD,C;MACb,eAA a,K;K;iEAGjB,Y;MAAsC,sBAAQ,C;K;+DAE9C,Y;MAAgC,mB;K;8DAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;Q AAoB,MAAM,6B;MAE1B,eAAO,mCAAP,EAAO,YAAP,C;MACA,OAAO,wBAAI,WAAJ,C;K;mEAGX,Y;MA AoC,sBAAQ,CAAR,I;K;+DAEpC,mB;MACI,wBAAI,YAAJ,EAAW,OAAx,C;MACA,mC;MACA,cAAO,E;K;+D AGX,mB;MIE1FJ,IAAI,EkEmFU,gBAAQ,ElEnFIB,CAAJ,C;QACI,ckEkFwB,4E;QIEjFxB,MAAM,6BAAsB,OAA Q,WAA9B,C;;MkEkFF,wBAAI,WAAJ,EAAU,OAAV,C;K;;IAIgb,+D;MAAuF,8B;MAAtF,kB;MAA0C,4B;MAC/ D,eAAyB,C;MAGrB,oCAAA,2BAAkB,gBAAIB,EAA6B,OAA7B,EAAcS,WAAK,KAA3C,C;MACb,eAAa,UAAU ,gBAAV,I;K;wDAGjB,0B;MACI,oCAAA,4BAAmB,KAAAnB,EAA0B,YAA1B,C;MAEb,WAAK,aAAI,mBAAY,K AAZ,IAAJ,EAAuB,OAAvB,C;MACL,mC;K;wDAGJ,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,YAAzB,C;MA Eb,OAAO,wBAAK,mBAAY,KAAZ,IAAL,C;K;6DAGX,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,YAAzB,C; MAEb,aAAa,WAAK,kBAAS,mBAAY,KAAZ,IAAT,C;MACIB,mC;MACA,OAAO,M;K;wDAGX,0B;MACI,oCA Aa,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,OAAO,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OAAvB,C;K ;MGAGO,Y;MAAQ,mB;K;2DAE/B,Y;MAA+C,WAAK,iB;K;;ICxMN,8B;MAAiC,sB;MAwCnF,uBAAoC,I;MA+ CpC,yBAA6C,I;K;IAIFR,oD;MAAC,wB;MAGIC,gBAAqB,K;K;iFAHa,Y;MAAA,yB;K;uGAKZ,Y;MAAQ,oB;K; 8DAE9B,oB;MAKI,eAAe,IAAK,S;MACpB,gBAAC,Q;MACd,OAAO,Q;K;wDAGX,Y;MAA+B,iEAAc,IAAd,C;K ;wDAC/B,Y;MAAkC,iEAAc,IAAd,C;K;sDACIC,iB;MAA4C,+DAAY,IAAZ,EAakB,KAAIB,C;K;;IAIB5C,8E;M AAA,wE;MAAsC,2CAAK,KAAM,IAAX,EAAGB,KAAM,MAAtB,C;MAAtC,Y;K;IASBJ,+C;MACsE,6B;K;mEA CIE,mB;MAAmD,kCAAc,OAAd,C;K;iEAEnD,mB;MAAiD,gCAAY,OAAZ,C;K;;yCAIrD,Y;MACI,YAAQ,Q;K;I AOQ,+F;MAAA,sD;MAAS,6B;K;uFACb,mB;MAAwC,MAAM,qCAA8B,8BAA9B,C;K;mFAC9C,Y;MACI,4BA AwB,Q;K;4FAG5B,mB;MAAsD,sDAAY,OAAZ,C;K;IAI3C,oH;MAAA,kD;K;4GACH,Y;MAAkC,OAAA,0BAA c,U;K;yGACHd,Y;MAAYB,OAAA,0BAAc,OAAO,I;K;2GAC9C,Y;MAAwB,0BAAc,S;K;;sFAL9C,Y;MACI,oBA AoB,oCAAQ,W;MAC5B,6G;K;0FAOJ,mB;MACI,qB;MACA,IAAI,+CAAY,OAAZ,CAAJ,C;QACI,4BAAwB,cA AO,OAAP,C;QACxB,OAAO,I;;MAEX,OAAO,K;K;oIAGY,Y;MAAQ,OAAA,4BAAwB,K;K;4FAEvD,Y;MAAsC ,4BAAwB,iB;K;;0FA9B1E,Y;MACI,IAAI,4BAAJ,C;QACI,6F;;MA+BJ,OAAO,mC;K;kDAKf,gB;MAEyB,Q;MA DrB,qB;MACqB,OAAA,I5EkR2D,QAAQ,W;M4EIRxF,OAAqB,cAArB,C;QAAqB,wB;QAaf,U5EmMsD,U;Q4E nMjD,Y5EgNiD,Y;Q4E/MxD,iBAAI,GAAJ,EAAS,KAAAT,C;;K;IAQc,iG;MAAA,sD;MAAS,oC;K;yFACf,mB;MA AwC,MAAM,qCAA8B,gCAA9B,C;K;qFAC9C,Y;MAAuB,4BAAwB,Q;K;8FAE/C,mB;MAAsD,wDAAc,OAAd, C;K;IAI3C,sH;MAAA,kD;K;8GACH,Y;MAAkC,OAAA,0BAAc,U;K;2GACHd,Y;MAAYB,OAAA,0BAAc,OAA O,M;K;6GAC9C,Y;MAAwB,0BAAc,S;K;;wFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5B,+G;K;sIAOmB,Y;M AAQ,OAAA,4BAAwB,K;K;8FAEvD,Y;MAAsC,4BAAwB,iB;K;;4FAnB1E,Y;MACI,IAAI,8BAAJ,C;QACI,iG;; MAoBJ,OAAO,qC;K;gDAGf,e;MACI,qB;MACA,WAaw,YAAQ,W;MACnB,OAAO,IAAK,UAAZ,C;QACI,YA AY,IAAK,O;QACjB,QAAQ,KAAM,I;QACd,IAAI,YAAO,CAAP,CAAJ,C;UACI,YAAY,KAAM,M;UACIB,IAA K,S;UACL,OAAO,K;;;MAGf,OAAO,I;K;kDAIX,Y;K;;IC3I+C,8B;MAAiC,oC;K;0CAEHf,iB;MAMI,IAAI,UAAU

,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,mCAAY,mBAAU,IAAV,EAAGB,KAaHb,C;K;4CAGvB,Y;MAG+B,OAAA,mCAAY,2BAAkB,IAAIB,C;K;;ICbT,0B;MAAuD,8B;MAAIC,4B;MACvD,4BAAkC,K;K;gCAkBIc,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;qCAGX,Y;K;iDAGA,uB;K;iFAG8B,Y;MAAQ,OAAA,oBAAM,O;K;sCAC5C,iB;MACyC,Q;MAAA,oCAAM,0BAAW,KAAX,CAAN,4D;K;sCACzC,0B;MAIW,IAAa,I;MAHpB,qB;MACA,0BAAW,KAAX,C;MAEoB,gBAAb,qBAAM,KAAN,C;MAAqB,qC;MAA5B,OAAO,CAAa,OIE8BjB,SkE9BI,2D;K;oCAGX,mB;MACI,qB;MACM,oBAAY,MAAK,OAAL,C;MACIB,qC;MACA,OAAO,I;K;sCAGX,0B;MACI,qB;MACM,oBAAY,QAAO,mCAAoB,KAAPB,CAAP,EAAMC,C;AAAnC,EAAsC,OAAtC,C;MACIB,qC;K;yCAGJ,oB;MACI,qB;MACA,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE/B,uBAAA,oBpEioDoB,QMhrD0C,Y8D+CrD,Q9D/CqD,CNgrD1C,C;MoEhoDpB,qC;MACA,OAAO,I;K;yCAGX,2B;MACI,qB;MACA,mCAAoB,KAAPB,C;MAEA,IAAI,UAAS,SAAb,C;QAAmB,OAAO,oBAAO,QAAP,C;MAC1B,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE3B,IADE,KACF,e;QAAQ,OAAO,oBAAO,QAAP,C;WACf,IAFE,KAef,O;QAAK,uB9D5DqD,Y8D4D7C,Q9D5D6C,CNgrD1C,QoEpnD6B,oBpEonD7B,C;;QoEnmDR,uBAAoC,cAA5B,oBAA4B,EAAY,CAAU,EAAP,KAAO,CAAY,Q9D7DE,Y8D6DK,Q9D7DL,C8D6DF,EAA4C,cAAN,oBAAM,EAAY,KAAZ,EAAMB,SAANB,CAA5C,C;;MAG5D,qC;MACA,OAAO,I;K;2CAGX,iB;MACI,qB;MACA,0BAAW,KAAX,C;MACA,qC;MACA,OAAW,UAAS,sBAAb,GACG,oBAAY,MADf,GAGG,oBAAY,QAAO,KAAP,EAAC,CAAd,CAAIB,CAAMC,CAAnC,C;K;uCAGR,mB;MAEkB,Q;MADd,qB;MACc,2B;MAAd,mD;QACI,IAAI,4BAAM,KAAN,GAAgB,OAaHb,CAAJ,C;UACU,oBAAY,QAAO,KAAP,EAAC,CAAd,C;UACIB,qC;UACA,OAAO,I;;MAGf,OAAO,K;K;8CAGX,8B;MACI,qB;MACA,qC;MACM,oBAAY,QAAO,SAAP,EAaKb,UAAU,SAAV,IAAIB,C;K;gCAGtB,Y;MACI,qB;MACA,uBhChHuC,E;MgCiHvC,qC;K;wCAIJ,mB;MAA+C,OAAm,QAAO,oBAAM,EAAY,QAAR,C;K;4CAErD,mB;MAAmD,OAAM,YAAN,oBAAM,EAAY,OAAZ,C;K;mCAEzD,Y;MAA0B,uBAAC,oBAAd,C;K;0CAE1B,iB;MAGe,UAGL,MAHK,EAMO,M;MAPIB,IAAI,KAAM,OAAN,GAAa,SAAJB,C;QACI,OAAO,2D;;MAGc,gBAAXb,eAAK,SAAL,IAAK,gBAAL,yB;MpEuwBL,UAAU,SAAV,EoEvwBsC,KpEuwBtC,EAD+F,CAC/F,EADoH,CACpH,EADuI,gBACvI,C;MoErwBI,IAAI,KAAM,OAAN,GAAa,SAAJB,C;QACI,MAAM,SAAN,IAAc,6E;;MAGIB,OAAO,K;K;kCAGX,Y;MACI,OAAO,EAAS,MAAM,MAAK,oBAAL,C;K;yCAI1B,Y;MACI,IAAI,yBAAJ,C;QAAGB,MAAM,oC;K;+CAG1B,iB;MACI,oCAAa,kCAAYB,SAAZB,C;MADoB,Y;K;wDAIrC,iB;MACI,oCAAa,mCAA0B,SAA1B,C;MAD6B,Y;K;;IAIJ9C,+B;MAAA,mD;MAG8B,sBhCRa,EgCQb,C;MAH9B,Y;K;IAKA,kD;MAAA,mD;MAIkD,sBhCdP,EgCcO,C;MAJID,Y;K;IAMA,2C;MAAA,mD;MAGqD,sB9DLA,Y8DKR,Q9DLQ,C8DKb,C;MAHrD,Y;K;ICrBJ,0C;MACI,IAAI,6BAAJ,C;QACU,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAawC,CAAxC,EAaiD,cAAN,KAAM,CAAjD,EAA4D,eAAW,UAAAX,CAA5D,C;;K;IAMiB,kD;MAAA,uB;QAAGB,OAAA,kBAAW,SAAQ,CAAR,EAAY,CAAX,C;O;K;IAFpD,4C;MACI,IAAI,6BAAJ,C;QACI,iBAaiB,gC;QACX,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAawC,CAAxC,EAaiD,cAAN,KAAM,CAAjD,EAA4D,UAA5D,C;;K;IAIR,gE;MACI,IAAI,aAAY,UAAU,CAAV,IAAZ,CAAJ,C;QACI,UAAU,KAAY,EAawC,SAAxC,EAAMD,UAAU,CAAV,IAAnD,EAAGe,UAAhE,C;;K;IAMiB,gC;MAAGB,OAAE,iBAAF,CAAE,EAAY,CAAV,C;K;IAF3C,0B;MACI,IAAI,6BAAJ,C;QACI,iBAaiB,gB;QACX,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAawC,CAAxC,EAaiD,cAAN,KAAM,CAAjD,EAA4D,cAA5D,C;;K;;IAAa,kD;MAAoB,QAAC,IAAM,CAAP,KAAa,IAAM,CAAnB,K;K;IARzC,uC;MACI,sC;QAAiC,OAAjC,yB;;MACA,4BAA4B,K;MAE5B,YAAY,E;MAGZ,iBAAC,CAAd,UAAaB,GAAtB,U;QAAiC,KAAY,MAAK,KAAL,C;MAC7C,iBAaiB,kC;MACX,KAAY,MAAK,UAAAL,C;MACIB,mBAAC,CAAd,YAAaB,KAAM,OAA5B,Y;QACI,QA AQ,MAAM,UAAQ,CAAR,IAAN,C;QACR,QAAQ,MAAM,OAAN,C;QACR,IAAI,CAAC,IAAM,CAAP,OAAc,IAAM,CAAPB,KAA0B,KAAK,CAAnC,C;UAAcC,OAAO,K;;MAEjD,4BAA4B,I;MAC5B,OAAO,I;K;IAIX,2D;MACI,aAAa,gBAAMB,KAAM,OAAzB,O;MACb,aAAa,YAAU,KAAY,EAaiB,MAAjB,EAAYB,KAazB,EAAGC,YAAhC,EAAS,C;MACb,IAAI,WAAW,KAAf,C;QACI,aAAU,KAAY,OAAiB,YAAjB,M;UAA+B,MAAM,CAAN,IAAW,OAAO,CAAP,C;;K;IAIID,4D;MAEI,IAAI,UAAS,GAAb,C;QACI,OAAO,K;;MAGX,aAAa,CAAC,QAAQ,GAAR,IAAD,IAAGB,CAAhB,I;MACb,WAAW,YAAU,KAAY,EAaiB,MAAjB,EAAYB,KAazB,EAAGC,MAAhC,EAawC,UAAxC,C;MACX,YAAY,YAAU,KAAY,EAaiB,MAAjB,EAAYB,SAAS,CAAT,IAAZB,EAAGqC,GAArC,EAAY,C;MAEZ,aAAiB,SAAS,MAAb,GAAqB,KAArB,GAAgC,M;MAG7C,gBAAGB,K;MAChB,iBAaiB,SAAS,CAAT,I;MACjB,aAAU,KAAY,OAAiB,GAAjB,M;QAEQ,iBAaA,MAAb,IAAuB,cAAc,GAArC,C;UACI,gBAAGB,KAAK,SAAL,C;UACHb,iBAaiB,MAAM,UAAN,C;UAEjB,IAAI,UAAW,SAAQ,SAAR,E

AAmB,UAAAnB,CAAX,IAA6C,CAAjD,C;YACI,OAAO,CAAP,IAAY,S;YACZ,6B;;YAEA,OAAO,CAAP,IAAY,  
U;YACZ,+B;;eAGR,iBAaA,MAAb,C;UACI,OAAO,CAAP,IAAY,KAACK,SAAL,C;UACZ,6B;;UAGA,OAAO,CA  
AP,IAAY,MAAM,UAAAN,C;UACZ,+B;;MAMZ,OAAO,M;K;ICrGX,4C;MAMoB,UACM,M;MAHtB,IAAI,iBA  
J,C;QAAkB,OAAO,C;MACzB,aAAa,C;MACb,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAEQ,oB;UAA  
mB,U;;UACnB,I5BFiC,MAAa,Y4BEnC,O5BFmC,C4BE9C,C;YAAwD,iCAAhC,OAAgC,C;iBAExD,uC;YAAmC  
,2BAAR,OAAQ,C;eACnC,wC;YAAmC,2BAAR,OAAQ,C;eACnC,sC;YAAmC,2BAAR,OAAQ,C;eACnC,uC;YA  
AmC,2BAAR,OAAQ,C;;YAEA,kBAAR,OAAQ,C;;QATvC,wB;QAYA,SAAS,MAAK,MAAL,QAAc,WAAAd,I;;M  
AEb,OAAO,M;K;;ICTP,uC;MAAA,2C;K;2DACI,0B;MAA2D,sBAAU,MAAV,C;K;gEAE3D,iB;MAA6C,Q;MA  
AA,wEAAqB,C;K;;IAHtE,mD;MAAA,kD;QAAA,iC;;MAAA,2C;K;;MC0BA,iC;MAKA,8B;MA6CA,0BAAMe,  
I;;IAzEnE,kC;MAAA,oB;MAA+B,8C;K;2CAE3B,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;uCAC/D,Y;MAC  
I,WAAa,Q;K;uDAGjB,mB;MAAgE,OAAA,WAAa,uBAAc,OAAc,C;K;0CAE7E,Y;MAAwE,OAAA,iCAAY,W;K  
;qDAEPf,mB;MACI,IAAI,iBAAS,OAAAT,CAAJ,C;QACI,WAAa,cAAO,OAAQ,IAAf,C;QACb,OAAO,I;;MAEX,  
OAAO,K;K;wFAGY,Y;MAAQ,OAAA,WAAa,K;K;;8BA6ChD,Y;MACI,0BAAY,Q;K;0CAIhB,e;MAAmD,OAA  
A,0BAAY,gBAAS,GAAT,C;K;4CAE/D,iB;MAAmE,gBAAZ,0B;MAAY,c;;QnEqnDnD,Q;QADhB,IAAI,wCAAs  
B,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,ImErnDm  
D,uBAAS,gBnEqnD9C,OmErnDwD,MAAV,QnEqnD5D,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MmEt  
nDgD,iB;K;kFAInD,Y;MACI,IAAI,+BAAJ,C;QACI,0BAAW,qB;;MAEf,OAAO,sC;K;uCAGf,Y;MAAgF,iC;K;kC  
AEhF,e;MAA+C,OAAA,0BAAY,WAAI,GAAJ,C;K;oCAE3D,sB;MAAgD,OAAA,0BAAY,aAAI,GAAJ,EAAS,K  
AAT,C;K;qCAE5D,e;MAAYC,OAAA,0BAAY,cAAO,GAAP,C;K;+EAEvB,Y;MAAQ,OAAA,0BAAY,K;K;;IA5D  
ID,0C;MAAA,iD;MAAuD,8B;MAvC3D,mB;MAwCQ,8BAAMb,W;MACnB,2BAAgB,WAA,Y,S;MAFhC,Y;K;IA  
KA,+B;MAAA,iD;MAGuB,aAAK,kEAAL,Q;MAHvB,Y;K;IAKA,4D;MAAA,iD;MAQ8D,qB;MzEpC9D,IAAI,Ey  
EsCQ,mBAAmB,CzEtC3B,CAAJ,C;QACI,cyEqCgC,+C;QzEpChC,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,I  
AAI,EyEuCQ,cAAc,CzEvCtB,CAAJ,C;QACI,gByEsC2B,yC;QzErC3B,MAAM,gCAAYB,SAAQ,WAAjC,C;;MyE  
0BV,Y;K;IAcA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAASB,GAATB,Q;MAA3C,Y;K;IAGA,yC;MAAA,iD;MAG  
8C,qB;MAC1C,KAACK,gBAAO,QAAP,C;MAJT,Y;K;IAqCJ,4B;MAK8E,gBAAnE,aAAmB,gEAAmB,C;MAA2E,  
wB;MAAIF,OtEvCO,S;K;;MuEjEP,uB;;kCAyCA,mB;MACI,UAAU,gBAAI,aAAI,OAAJ,EAaA,IAAb,C;MACd,O  
AAO,W;K;8BAGX,Y;MACI,gBAAI,Q;K;uCAOR,mB;MAA6D,OAAA,gBAAI,mBAAY,OAAZ,C;K;gCAEjE,Y;  
MAAYC,OAAA,gBAAI,U;K;iCAE7C,Y;MAAqD,OAAA,gBAAI,KAACK,W;K;qCAE9D,mB;MAAkD,OAAA,gB  
AAI,cAAO,OAAP,CAAJ,Q;K;+EAEpB,Y;MAAQ,OAAA,gBAAI,K;K;;IA5D1C,6B;MAAA,iD;MAGoB,8B;MAZ  
xB,mB;MAAq,oBAAM,gB;MAJV,Y;K;IAOA,yC;MAAA,iD;MAG2C,8B;MANb/C,mB;MAoBQ,oBAAM,eAAgB  
,QAAS,KAAzB,C;MACN,qBAAO,QAAP,C;MALJ,Y;K;IAQA,4D;MAAA,iD;MAQ2D,8B;MAhC/D,mB;MAiCQ,  
oBAAM,eAAgB,eAAhB,EAAiC,UAAjC,C;MATV,Y;K;IAYA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAASB,GA  
AtB,Q;MAA3C,Y;K;IAEA,oC;MAAA,iD;MAM0C,8B;MA5C9C,mB;MA6CQ,oBAAW,G;MAPf,Y;K;IAmCJ,+B;  
MAKuC,gBAA5B,eAAQ,eAAR,C;MAAoC,6B;MAA3C,OvENO,S;K;IwEzD6B,uC;MAAC,kC;MAErC,oBAAkC,  
kB;MACiC,sBAAYB,C;K;2EAHY,Y;MAAA,8B;K;2FAGrC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;gDAGA,sB  
;MACI,eAAe,aAAS,qBAAY,GAAZ,C;MACxB,mBAAMb,6BAASB,QAATB,C;MACnB,IAAI,oBAAJ,C;QAEI,kB  
AAW,QAAX,IAAuB,mCAAY,GAAZ,EAAiB,KAAjB,C;;QAEvB,IAAI,6BAAJ,C;UAEI,YAA+B,Y;UAC/B,IAAI,  
aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,CAAb,C;YACI,OAAO,KAAM,gBAAS,KAAT,C;;YAEb,kBAAW,  
QAAX,IAAuB,CAAQ,KAAR,EAAe,mCAAY,GAAZ,EAAiB,KAAjB,CAAf,C;YACvB,6B;YACA,OAAO,I;;UAI  
X,YAAuC,Y;UACvC,cAAkB,wBAAN,KAAM,EAAiB,GAAjB,C;UACIB,IAAI,eAAJ,C;YACI,OAAO,OAAM,gB  
AAS,KAAT,C;;UAEX,KAAY,MAAK,mCAAY,GAAZ,EAAiB,KAAjB,CAAL,C;;MAG1B,6B;MAEA,OAAO,I;  
K;iDAGX,e;MAE0D,IAAnC,I;MADnB,eAAe,aAAS,qBAAY,GAAZ,C;MACL,oCAASB,QAATB,C;MAAA,iB;QA  
AmC,OAAO,I;;MAA7D,mBAAMb,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;QAChC,IAAI,aAAS,gBAAO,KA  
AM,IAAb,EAakB,GAAIB,CAAb,C;U9BzDR,O8B0D6B,iB9B1DhB,C8B0D4B,Q9B1D5B,C;U8B2DD,6B;UACA  
,OAAO,KAAM,M;;UAEb,OAAO,I;;QAGX,YAAuC,Y;QACvC,8BAAC,KAAAd,iB;UACI,cAAY,MAAM,KAAN,C  
;UACZ,IAAI,aAAS,gBAAO,GAAP,EAAY,OAAM,IAAIB,CAAb,C;YACI,IAAI,KAAM,OAAN,KAAc,CAAIB,C;  
cACU,KAAN,UAA2B,C;c9BtE/C,O8BwEqC,iB9BxEb,C8BwEoC,Q9BxEpC,C;;c8B2Ea,KAAY,QAAO,KAAP,  
EAAc,CAAd,C;;YAEtB,6B;YAEA,OAAO,OAAM,M;;MAIzB,OAAO,I;K;0CAGX,Y;MACI,oBAaA,kB;MACb,

YAAO,C;K;mDAGX,e;MAAyC,uBAAS,GAAT,S;K;8CAEzC,e;MAA+B,Q;MAAA,+BAAS,GAAT,8B;K;+CAE/  
B,e;MAC2E,IAApD,I;MAAA,oCAAsB,aAAS,qBAAY,GAAZ,CAA/B,C;MAAA,iB;QAAoD,OAAO,I;MAA9E,m  
BAAmB,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;QACHC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,  
CAAb,C;UACI,OAAO,K;UAEP,OAAO,I;QAGX,YAAuC,Y;QACvC,OAAa,wBAAN,KAAM,EAaiB,GAAjB,C  
;K;uDAIrB,0B;MACI,sB;Q1FsoCY,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,I0FtoCK,aAAS,gB1FsoCA,00FtoC  
a,IAAb,M1FsoCd,C;YAAwB,qBAAO,O;YAAP,uB;QAC9C,qBAAO,I;M0FvoCH,yB;K;IAIO,8E;MAAA,wD;M  
ACH,aAAY,E;MAEZ,YAA0B,MAAa,MAAK,qCAAL,C;MACvC,gBAAe,E;MAEf,oBAA4B,I;MAC5B,eAAc,K;  
MACd,iBAAgB,E;MACHb,iBAAqC,I;K;yEAERc,Y;MACI,IAAI,6BAAwB,YAA5B,C;QACI,gBAAqB,iBAAqD,  
O;QAC1E,IAAI,4DAAc,SAaIB,C;UACI,OAAO,C;MAGf,IAAI,yDAAa,SAAK,OAAtB,C;QACI,oBAAe,2CAA  
W,UAAK,aAAL,CAAX,C;QACf,eAAU,iC;QACV,iBAAY,C;QACZ,OAAO,C;QAEp,oBAAe,I;QACf,OAAO,C;  
K;mEAIf,Y;MACI,IAAI,eAAS,EAAb,C;QACI,aAAQ,oB;MACZ,OAAO,eAAS,C;K;gEAGpB,Y;MAEoB,Q;MAD  
hB,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACN,IAAI,YAAJ,C;QACZ,yBAAqD,cAARd,C;QAEa,OAAb,iB;  
;MAHJ,oB;MAKA,iBAaiB,S;MACjB,aAAQ,E;MACR,OAAO,S;K;KEAGX,Y;M3E/CR,I2EgDyB,c3EhDrB,QAA  
J,C;QACI,cAhByB,0B;QaiBzB,MAAM,6BAAsB,OAAQ,WAA9B,C;M2E+CE,6BAAYB,cAAO,6BAA,Y,IAAnB,  
C;MACzB,iBAAY,I;MAEZ,uC;K;6CAtdZ,Y;MAEI,2D;K;4DAyDJ,oB;MACI,mBAAmB,kBAAW,QAAX,C;MA  
CnB,OAAW,iBAaiB,SAArB,GAAgC,IAAhC,GAA0C,Y;K;wCCtKrD,Y;MACI,aAAgB,MAAM,OAAI,CAaiB  
,IAAjB,C;MAEb,OAAO,KAAP,IAAgB,C;M/BXpB,O+BYqB,M/BZR,C+BYgB,K/BZhB,C;M+BaT,OAAO,M;K;  
ICNuB,qC;MAAC,kC;MAEnC,oBAakC,kB;MACIC,sBAAYB,C;K;yEAHU,Y;MAAA,8B;K;yFAGnC,Y;MAAA,0  
B;K,OAAA,gB;MAAA,0B;K;iDAWA,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,K;MAC3B,OAAO,kBAAW,GAA  
X,MAAoB,S;K;4CAG/B,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,I;MAC3B,YAA,Y,kBAAW,GAA,X,C;MACZ,O  
AAW,UAAU,SAArB,GAAgC,KAAhC,GAA2D,I;K;8CAI/D,sB;M7EVA,IAAI,E6EWQ,uB7EXR,CAAJ,C;QACI,c  
Ada,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;M6EUN,eAAe,kBAAW,GAA,X,C;MACf,kBAAW,GAA,X,IA  
AkB,K;MAEIB,IAAI,aAAa,SAAjB,C;QACI,6B;QAEA,OAAO,I;QAGP,OAAO,Q;K;+CAIf,e;MACI,IAAI,0BAA  
J,C;QAAoB,OAAO,I;MAC3B,YAA,Y,kBAAW,GAA,X,C;MACZ,IAAI,UAAU,SAAd,C;QhCnDJ,OgCoDyB,iBhC  
pDZ,CgCoDwB,GhCpDxB,C;QgCqDL,6B;QAEA,OAAO,K;QAGP,OAAO,I;K;wCAKf,Y;MACI,oBAAa,kB;MA  
Cb,YAAO,C;K;IAKA,0E;MAAA,oD;MACH,cAAkC,MAAa,MAAK,mCAAL,C;MAC/C,kBAA4B,qBAAL,WAA  
K,C;MAC5B,iBAA+B,I;K;iEAE/B,Y;MAAkC,OAAA,eAAS,U;K;8DAE3C,Y;MAIuB,gB;MAHnB,UAAU,eAAS,  
O;MACnB,iBAAU,G;MAES,+E;MAAnB,OAAO,iD;K;gEAGX,Y;MAEkC,UAA9B,M;MAAA,oC;MAA8B,YAAa  
,c;M7EchD,uB;MAeP,IAfoB,KAehB,QAaj,C;QACI,cAhByB,0B;QaiBzB,MAAM,6BAAsB,OAAQ,WAA9B,C;  
QAEN,sBAnBgB,K;M6Ede,oBAAO,sFAAP,C;K;2CAjBnC,Y;MACI,yD;K;IAqBkd,0F;MAAA,8B;MAAA,oD;K  
;KHAC9B,Y;MAAQ,uB;K;oHACN,Y;MAAQ,6CAAuB,gBAAvB,C;K;2EAE9B,oB;MAAwC,OAAA,2BAAuB,aA  
AI,gBAAJ,EAAS,QAAT,C;K;qEAE/D,Y;MAA+B,OAAA,mCAAY,uBAAc,IAAd,C;K;qEAC3C,Y;MAAkC,OAA  
A,mCAAY,uBAAc,IAAd,C;K;mEAC9C,iB;MAA4C,OAAA,mCAAY,qBAAY,IAAZ,EAakB,KAAIB,C;K;gDAR  
5D,e;MAAsD,iE;K;MCItd,sBAOsC,I;MA6CtC,yB;MAOA,4BAakC,K;IArIE,sD;MAZpC,oB;MAYyD,0CAAq  
C,GAArC,EAA0C,KAA1C,C;MACrD,oBAAuC,I;MACvC,oBAAuC,I;K;wDAEvC,oB;MACI,WAAmB,iB;MACn  
B,OAAa,mEAAS,QAAT,C;K;IAIrB,wC;MAAA,oB;MAA+B,8C;K;IAE3B,sD;MAAA,oB;MACI,cAcS,C,I;MAEt  
C,cAcS,C,I;MAGlC,cAAO,iC;K;6DAIX,Y;MACI,OAAO,gBAAS,I;K;0DAGpB,Y;MAEI,IAAI,CAAC,cAAL,C;Q  
AAgB,MAAM,6B;MAEtB,cAAc,0B;MACd,cAAO,O;MACa,gBAAb,OAAQ,a;MAAf,c3E0DS,S2E1D0B,KAAO,  
iC3E0DzC,GAAqB,SAArB,GAA+B,I;M2EzD1B,OAAO,O;K;4DAGX,Y;M9EwBR,IAAI,E8EvBc,eAAQ,I9EuBtB  
,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,WAA9B,C;M8ExBE,WAAc,iB;MAGP,oCAAP,0BAA  
O,C;MACP,gCAAI,cAAO,0BAAO,IAAd,C;MAEJ,cAAO,I;K;iDAIf,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;  
K;6CAC/D,Y;MACI,WAAmB,Q;K;6DAGvB,mB;MAAgE,OAAA,WAAmB,uBAAc,OAAc,C;K;gDAEnF,Y;MA  
AwE,qD;K;2DAExE,mB;MACI,qB;MACA,IAAI,iBAAS,OAAI,CAAJ,C;QACI,WAAmB,cAAO,OAAQ,IAAf,C;  
QACnB,OAAO,I;MAEX,OAAO,K;K;8FAGY,Y;MAAQ,OAAA,WAAmB,K;K;sDAEID,Y;MAAsC,WAAmB,iB;  
K;iDAa7D,qB;M9ErBA,IAAI,E8E0BM,0BAAQ,IAAR,IAAgB,0BAAQ,I9E1B9B,CAAJ,C;QACI,cAdW,e;QAeX,  
MAAM,6BAAsB,OAAQ,WAA9B,C;M8E0BN,YAA,Y,mB;MACZ,IAAI,SAAS,IAAb,C;QACI,sBAAO,S;QACP,y  
BAAO,S;QACP,yBAAO,S;QAGK,YAAa,KAAM,a;Q9EIBhC,uB;QAeP,IAfoB,KAehB,QAaj,C;UACI,gBAhByB  
,0B;UaiBzB,MAAM,6BAAsB,SAAQ,WAA9B,C;UAEN,sBAnBgB,K;Q8EkBZ,+B;QAEA,yBAAO,K;QACP,yB

AAO,K;QAEP,qBAaA,S;QACb,qBAaA,S;;K;+CAIrB,qB;MAII,IAAI,SAAK,aAAL,KAAc,SAaIB,C;QAEL,sBAA  
O,I;;QAEP,IAAI,wBAAS,SAAb,C;UAEL,sBAAO,sB;;QAEX,qDAAc,sB;QACd,qDAAc,sB;;MAEIB,yBAAO,I;M  
ACP,yBAAO,I;K;oCA8CX,Y;MAEL,qB;MACA,4BAaA,I;MACb,OAAO,I;K;oCAGX,Y;MACI,qB;MACA,kBAAI  
,Q;MACJ,sBAAO,I;K;gDASX,e;MAAmD,OAAA,kBAAI,mBAAY,GAaZ,C;K;kDAEvD,iB;MACyC,IAAR,I;MA  
AA,0B;MAAA,iB;QAAQ,OAAO,K;;MAA5C,WAA6B,I;;QAEzB,IAAI,OAAA,IAAK,MAAL,EAaC,KAAAd,CAAJ  
,C;UACI,OAAO,I;;QAEX,OAAO,cAAA,IAAK,aAAL,C;;MACF,iBAAS,mBAAT,C;MACT,OAAO,K;K;6CAIX,  
Y;MAAoF,uC;K;wCAEpF,e;MAAmD,Q;MAAJ,QAAL,OAAJ,kBAAI,WAAI,GAaJ,CAAJ,6B;K;0CAE/C,sB;MA  
CI,qB;MAEA,UAAU,kBAAI,WAAI,GAaJ,C;MACd,IAAI,OAAO,IAAX,C;QACI,eAAe,mCAAW,GAAX,EAAG  
B,KAAhB,C;QACf,kBAAI,aAAI,GAaJ,EAAS,QAAT,C;QACK,wBAAT,QAAS,C;QACT,OAAO,I;;QAEP,OAA  
O,GAaI,gBAAS,KAAT,C;;K;2CAInB,e;MACI,qB;MAEA,YAAy,kBAAI,cAAO,GAAP,C;MACHb,IAAI,SAAS,I  
AAb,C;QACU,sBAAN,KAAM,C;QACN,OAAO,KAAM,M;;MAEjB,OAAO,I;K;qFAGmB,Y;MAAQ,OAAA,kBA  
AI,K;K;6CAE1C,Y;MACI,IAAI,yBAAJ,C;QAAGB,MAAM,oC;K;;IAng1B,mC;MAAA,uD;MAGuB,qB;MA9J3B,  
yB;MA+JQ,sBAAM,gB;MAJV,Y;K;IAOA,iD;MAAA,uD;MAAoD,qB;MAIKxD,yB;MAoKc,Q;MAAN,sBAAM,+  
D;MAFV,Y;K;IAKA,kE;MAAA,uD;MAQ8D,eAAM,eAAN,EAaU,B,UAAvB,Q;MA/KIE,yB;MAGLQ,sBAAM,gB  
;MATV,Y;K;IAYA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAASB,GAAtB,Q;MAA3C,Y;K;IAEA,+C;MAAA,uD  
;MAG2C,qB;MAxL/C,yB;MAyLQ,sBAAM,gB;MACN,KAAK,gBAAO,QAAP,C;MALT,Y;K;IA6EJ,kC;MAKwD  
,gBAA7C,qBAAYB,eAAzB,C;MAAqD,wB;MAA5D,O3EjMO,S;K;;oC4EvCP,Y;MAEK,Q;MAA8B,CAA9B,2EA  
A8B,S;MAC/B,OAAO,I;K;6CAGX,Y;MAA+C,gBAAI,iB;K;;IAhCnD,wC;MAAA,uD;MAAmD,eAAM,GAAN,Q;  
MAPvD,yB;MAOI,Y;K;IAEA,qC;MAAA,uD;MAGuB,eAAM,oBAAN,Q;MAZ3B,yB;MASI,Y;K;IAKA,+C;MAA  
A,uD;MAG8C,eAAM,oBAAN,Q;MAjBID,yB;MAkBQ,qBAAO,QAAP,C;MAJJ,Y;K;IAOA,kE;MAAA,uD;MAQ8  
D,eAAM,qBAASB,eAAtB,EAaU,C,UAAvC,CAAN,Q;MA7BIE,yB;MAqBI,Y;K;IAUA,sD;MAAA,uD;MAA2C,qB  
AAK,eAAL,EAASB,GAAtB,Q;MAA3C,Y;K;IAGBJ,qC;MAKMD,gBAAXC,mBAAC,qBAAd,C;MAAGD,6B;MAA  
vD,O5EoBO,S;K;;kF6EzEX,uB;MAQL,OAAO,O;K;ICXX,sB;K;mCACI,Y;MACI,mBAAM,IAAN,C;K;2CAGJ,  
mB;MACI,mBAAM,OAAN,C;MACA,c;K;;iCAKJ,Y;K;;IAKuB,oC;MAA8B,qB;MAA7B,gC;K;2CACxB,mB;MA  
EI,oBA+DyC,OA/Dd,OA+Dc,C;MA9DzC,iBAaA,OAAM,aAAN,C;K;;IAIrB,8B;MAEoC,qB;K;iDAChC,mB;MA  
CI,OAAQ,KAAI,OAAJ,C;K;mDAGZ,mB;MACI,OAAQ,KAAI,OAAJ,C;K;2CAGZ,Y;MACI,OAAQ,KAAI,EAaJ  
,C;K;;IAIhB,0B;MAEqC,qB;MACjC,cAAa,E;K;6CAEb,mB;MACI,eAoCyC,OApcxB,OAoCwB,C;K;qCAjC7C,Y  
;MACI,cAAS,E;K;;IAIjB,sC;MAE4C,yB;K;yDACxC,mB;MACI,QAwbYc,OAxB1B,OAwb0B,C;MAvBzC,QAA  
Q,CpEqJoF,aoErJhE,IpEqJgE,EoErJ1D,CpEqJ0D,C;MoEpJ5F,IAAI,KAAK,CAAT,C;QACI,4BAAU,CpEwL0E,W  
oExL9D,CpEwL8D,EoExL3D,CpEwL2D,C;QoEvLpF,Y;QACA,IAAI,CpEmLiE,WoEnLrD,IAAI,CAAJ,IpEmLqD  
,C;;MoEjLzE,4BAAU,C;K;iDAGd,Y;MACI,OAAQ,KAAI,WAAJ,C;MACR,cAAS,E;K;;IAWjB,yB;MACiD,cAA  
a,KAAb,C;K;IAEjD,mB;MAEL,MAAO,U;K;IAGX,4B;MAEL,MAAO,iBAAQ,OAAR,C;K;IAGX,wB;MAEL,MAA  
O,eAAM,OAAN,C;K;IAGX,kB;MACqC,MAAM,qCAA8B,sCAA9B,C;K;IAE3C,wB;MAC4C,MAAM,qCAA8B,4  
CAA9B,C;K;ICIGID,mD;MACI,0B;MASA,gBAA2B,a;K;2FAFvB,Y;MAAQ,OAAA,eAAS,Q;K;oDAIrB,kB;MA  
CI,UAAU,IAAK,S;MAEX,YAAQ,2CAAR,C;QACI,gBAAc,MAAO,M;WAEzB,YAAQ,yBAAR,C;QACI,gBAAc,  
yC;QACd,eAAS,oBAAW,MAAX,C;;QAEL,MAAM,6BAASB,iBAAtB,C;K;4CAItB,Y;MAOW,Q;MALP,IAAI,kB  
AAW,2CAAf,C;QACI,gBAAS,yB;QACT,OAAO,yB;;MAEX,aAAa,IAAK,S;MAEd,eAAW,yCAAX,C;QAASB,gC  
;WACtB,0C;QAA4B,MAAM,MAAO,U;;QACjC,a;MAHZ,W;K;;IA7BJ,gD;MAAA,0D;MACyD,6BAAK,QAAL,E  
AAe,2CAAf,C;MADzD,Y;K;;;ICRA,2C;MAAA,+D;MAAuB,iC;MAF3B,iC;MAEL,Y;K;IACA,sD;MAAA,+D;M  
AAuC,6BAAM,OAAN,Q;MAH3C,iC;MAGI,Y;K;IACA,6D;MAAA,+D;MAAmD,kCAAM,OAAN,EAaE,KAAf,  
C;MAJvD,iC;MAII,Y;K;IACA,oD;MAAA,+D;MAAiC,6BAAM,KAAN,Q;MALrC,iC;MAKI,Y;K;I1C4CJ,yE;MA  
SI,sC;MAAA,4C;K;IATJ,iGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,0DAaQ,kB;MACI,wBAAW,MAAX,C;K;IAdZ,  
sF;I2C5C2E,0C;M5CkKhE,Q;MADP,e4ChKA,M5CgKA,C;MACO,Q4CjKP,M5CiKO,+D;M4ChKX,W;K;;+FCuH  
A,gB;MACI,aAAa,IAAO,MAAP,E;MACb,KAAK,MAAL,C;MACA,OAAO,M;K;wFC3HX,yB;MAAA,uD;MAA  
A,wC;QAWqG,OAaK,cAAL,SAAK,EAaIB,IAAjB,EAaUB,IAAvB,C;O;KAXIG,C;wFAaA,yB;MAAA,uD;MAA  
A,wC;QAWoG,OAaK,cAAL,SAAK,EAaIB,IAAjB,EAaUB,IAAvB,C;O;KAXzG,C;8ECbA,yB;MAAA,6C;MAA  
A,sC;QAOyD,OAaK,SAAL,SAAK,EAAY,QAaZ,C;O;KAP9D,C;8EASA,yB;MAAA,6C;MAAA,wC;QAWkE,O  
AAK,SAAL,SAAK,EAaA,UAAb,S;O;KAXvE,C;oFAaA,yB;MAAA,mD;MAAA,wC;QAWqE,OAaK,YAAL,SA

AK,EAAGB,UAAhB,S;O;KAX1E,C;kFCZI,yB;MAAA,iD;MAAA,4B;QAAe,OAAK,WAAL,SAAK,C;O;KAApB,C;wFAYA,yB;MAAA,uD;MAAA,4B;QAAe,OAAK,cAAL,SAAK,C;O;KAApB,C;IC5BJ,gC;MAAoE,gCAAqB,OAArB,C;K;IAEIC,uC;MAAC,wB;K;iDAC/B,iB;MACI,eAAQ,KAAR,C;K;8CAGJ,Y;MAAyC,iCAAuB,cAAvB,M;K;;ICCO,6C;MAAA,8B;MAAS,uB;K;8FACIC,Y;MAAQ,OAAA,gBAAY,O;K;mDAE3C,iB;MACI,IADoC,KACpC,IAAG,CAAH,IADoC,KACpC,IAAM,sBAAN,C;QAD8B,OACX,gBAAY,MAAK,KAAL,C;;QACvB,MAAM,8BAA0B,WAAQ,KAAR,6BAAmC,sBAAnC,MAA1B,C;K;;IARtB,8B;MAGoD,4C;K;wECFpD,yB;MAAA,uC;MAAA,4B;QAOsC,MAAL,SAAK,C;O;KAPtC,C;kFASA,yB;MAAA,iD;MAAA,kC;QAWuD,OAAK,WAAL,SAAK,EAAC,IAAd,C;O;KAX5D,C;+ECfA,qB;MAI8C,gB;K;iFAE9C,qB;MAIsE,OAAK,S;K;kFAE3E,qB;MAMyE,gB;K;IAEzE,6B;MAiBa,UAPF,M;MAFP,QAAc,S;MAGV,cAAK,UAAU,U;QACI,mBAAK,UAAU,G;;QACJ,IjDzBqC,MAAa,YiDyBvC,CjDzBuC,CiDyBID,C;UAC6B,8BAAZB,CAAyB,C;;UAGN,UAAIB,uDAaKB,Y;;MAP3B,a;K;IC7B6D,gD;K;;ICDjE,2B;MAEI,MAAM,yBAAqB,OAArB,C;K;IAGV,sB;MAEI,MAAM,uBAAMb,cAAnB,C;K;IAGV,2B;MAEI,MAAM,6BAAsB,OAAtB,C;K;IAGV,iC;MAEI,MAAM,4CAAqC,uBAAqB,YAArB,8BAArC,C;K;ICIBV,8B;MC8CW,kBzGqBiD,oB;MyGM9C,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAaKB,sBAAY,GAAZ,C;QAKFiD,U;QAJFnE,WzGyKJ,ayGzKgB,GzGyKhB,EwG50oB,CCmEkC,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAiFhD,GDpJrC,CCoJqC,GAA6B,UAJfJc,WaifiC,6DDpJnD,IAAM,CAAN,IxG4OpB,C;;MwG5OA,OCqEO,W;K;IC3EyB,oC;MAAC,oC;K;;;iF/CFrC,kD;MAyDI,SAAy,MAAK,OAAL,EAAC,SAAd,EAAYB,OAazB,C;K;iFCzDhB,iC;MAuBmC,0B;QAAA,aAAuD,S;MACTf,SAAY,MAAK,UAAU,C;K;;;I+CoBhB,qB;MAK0B,Q;MADtB,UAAmB,E;MACnB,wBAAsB,KAAtB,gB;QAAsB,aAAA,KAAtB,M;QAAK,IAAC,0BAAD,EAAO,2B;QACR,IAAI,IAAJ,IAAY,K;;MAEhB,OOAO,G;K;IAGX,+B;MAMgB,Q;MADZ,WAA0B,MAAa,MAAK,KAAL,C;MACvC,wBAAY,IAAZ,gB;QAAY,UAAA,IAAZ,M;QACI,IAAU,KAAY,gBAAe,GAAf,CAAtB,C;UACI,UAAK,GAAL,IAAY,MAAM,GAAN,C;;MAGpB,OOAO,S;K;qEC5DX,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOKD,OAAA,MAAW,KAAL,CAAJ,C;O;KAP7D,C;uEASA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;yEAWA,yB;MAAA,iB;MAAA,uB;QAKb+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAIB1E,C;uEAoBA,yB;MAAA,0B;MAAA,oB;QAUmD,kBAAW,CAAX,C;O;KAVnD,C;uEAYA,yB;MAAA,0B;MAAA,oB;QASmD,kBAAW,CAAX,C;O;KATnD,C;uEAWA,yB;MAAA,0B;MAAA,oB;QAUmD,kBAAW,CAAX,C;O;KAVnD,C;yEAYA,yB;MAAA,4B;MAAA,oB;QAYoD,mBAAY,CAAZ,C;O;KAZpD,C;yEAca,yB;MAAA,4B;MAAA,oB;QAYoD,mBAAY,CAAZ,C;O;KAZpD,C;yEAca,yB;MAAA,4B;MAAA,oB;QAaoD,mBAAY,CAAZ,C;O;KAbpD,C;yEAea,yB;MAAA,4B;MAAA,uB;QAS+D,mBAAY,CAAZ,EAae,CAaf,C;O;KAT/D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAQmD,OAAA,MAAW,MAAK,CAAL,C;O;KAR9D,C;qEAUA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAAW,KAAL,CAAJ,C;O;KAV7D,C;yEAYA,yB;MAAA,4B;MAAA,oB;QAcoD,mBAAY,CAAZ,C;O;KAdpD,C;IagBA,sB;MAcI,IAAI,QAAQ,GAAR,IAAE,SAAQ,GAA3B,C;QAAgC,OOAO,wCAA0,I;MAC9C,OOAO,IAAW,KAAL,CAAJ,CAAX,GAAoB,IAAW,KAAL,IAAJ,C;K;mEAG1C,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAAW,KAAL,CAAJ,C;O;KAX5D,C;yEAaA,yB;MAAA,4B;MAAA,oB;QAoD,mBAAY,CAAZ,C;O;KAPpD,C;uEASA,yB;MAAA,0B;MAAA,oB;QAOMD,kBAAW,CAAX,C;O;KAPnD,C;uEASA,yB;MAAA,4B;MAAA,oB;QAgBmD,mBAAY,CAAZ,C;O;KAhBnD,C;uEAkBA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAUoD,OAAA,MAAW,OAAM,CAAN,C;O;KAV/D,C;+EAYA,yB;MAAA,4B;MAAA,oB;QAUuD,mBAAY,CAAZ,C;O;KAVvD,C;IAYA,kB;MAQI,IAAI,IAAI,GAAJ,KAAW,GAAf,C;QACI,OOAO,IAAW,OAAM,CAAN,C;;MAEtB,YAzBgD,MAAW,OAYBzC,CAZByC,C;MA0B3D,OOAW,QAAQ,CAAR,KAAA,GAAxB,GAA6B,KAA7B,GAtC+C,MAAW,MAcB,CAtCa,C;K;qEAyC9D,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAAW,KAAL,CAAJ,C;O;KAV7D,C;uEAYA,yB;MAAA,0B;MAAA,oB;QAWmD,kBAAW,CAAX,C;O;KAXnD,C;wEAca,yB;MAAA,iB;MAAA,uB;QA06D,OAAA,MAAW,KAAL,CAAJ,EAAO,CAAP,C;O;KAPxE,C;wEASA,yB;MAAA,iB;MAAA,uB;QA06D,OAAA,MAAW,KAAL,CAAJ,EAAO,CAAP,C;O;KAPxE,C;uEAUA,yB;MAAA,iB;MAAA,oB;QAamD,OAAA,MAAW,MAAK,CAAL,C;O;KAb9D,C;qEAkBA,yB;MAAA,iB;MAAA,+B;QAayD,OAAA,MAAW,KAAL,SAAJ,EAAU,CAAV,C;O;KAbpE,C;uEAea,yB;MAAA,iB;MAAA,+B;QAOSD,O

AAA,MAAW,KAAI,SAAJ,EAAY,CAAZ,C;O;KAPjE,C;iGAmBsD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,KAAI,SAAJ,C;O;KAAAnB,C;+EaAT,yB;MAAA,0B;MAAA,4B;QAAQ,kBAAW,SAAX,C;O;KAAR,C;iFAE7C,yB;MAAA,6C;MAAA,kC;QAK8D,OAAK,SAAL,SAAK,EAAC,IAAD,C;O;KALnE,C;IAkBqC,4B;MACjC,gBA AO,CAAP,C;QADyC,OACrB,QAAP,CAAC,SAAM,C;WACpB,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCA AO,kBAA/B,C;QAFyC,OA EW,S;WACpD,kBAAQ,wCAAO,UAAf,C;QAHyC,OAGb,YAAY,SAAL,SAAK,C;;Q AHC,OAI5B,OAAL,SAAK,CAAL,GAAgB,S;K;IAG5B,2B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wC AAO,kBAA/B,C;QADwC,OACY,S;WACpD,kBAAQ,GAAR,C;QAFwC,OA EZB,wCAAO,U;;QACP,WAAc,UAA L,SAAK,CAAL,yBA AuB,YAAO,CAAX,GAAc,CAAd,GAAqB,EAAXC,E;QAHgB,OpDjc6B,MAAa,gBA Ae,IAA f,C;;K;IoDuctF,6B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAAO,kBAA/B,C;QAD0C,OACU,S;WA CpD,kBAAQ,GAAR,C;QAF0C,OA E3B,CAAC,wCAAO,U;;QACR,WAAc,UAA L,SAAK,CAAL,yBA AuB,YAAO ,CAAX,GAAc,EAAd,GAA sB,CAAzC,E;QAHkB,OpD3c2B,MAAa,gBA Ae,IAAf,C;;K;IoDkdtF,oC;MAUI,IAAK, QAAL,SAAK,CAAL,IAAmB,QAAH,EAAG,CAAnB,C;QADuD,OACzB,wCAAO,I;WACrC,WAA M,SAAN,C;Q AFuD,OA EZC,E;WACd,SAAK,SAAL,C;QAHuD,OAGrC,OAAL,SAAK,C;;QAHqC,OAI1B,SAAL,SAAK,C;K;I AIjC,+B;MAYI,uB;QAAW,MAAM,gCAAyB,yBA AzB,C;WACjB,gBA AO,UAA P,C;QAFyC,OA E jB,U;WACxB, gBA AO,WAA P,C;QAHyC,OAGjB,W;;QAHiB,OAI V,YAA vB,IAAW,OAAM,SAAN,CAAY,C;K;IAGnC,gC;MA YI,uB;QAAW,MAAM,gCAAyB,yBA AzB,C;WACjB,oD;QAF2C,+B;WAG3C,oD;QAH2C,+B;;QAAA,OAI Z,uB AA vB,IAAW,OAAM,SAAN,CAAY,C;K;uEASnC,yB;MAAA,iB;MAAA,oB;QAOGD,OAAA,MAA6B,KAAZ,CA AY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOGD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;yEASA,yB;MA AA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA, MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY ,C;O;KAT/E,C;2EAWA,yB;MAAA,iB;MAAA,uB;QAKB4D,OAAA,MAA6C,OAA1B,CAA0B,EA AZ,CAAY,C;O ;KAI BzG,C;yEAoBA,yB;MAAA,0B;MAAA,oB;QAUiD,OAAyB,WAAZ,CAAY,C;O;KAV1E,C;yEAYA,yB;MA AA,0B;MAAA,oB;QASiD,OAAyB,WAAZ,CAAY,C;O;KAT1E,C;yEAWA,yB;MAAA,0B;MAAA,oB;QAUiD,O AAyB,WAAZ,CAAY,C;O;KAV1E,C;2EAYA,yB;MAAA,4B;MAAA,oB;QAYkD,OAA0B,YAAZ,CAAY,C;O;K AZ5E,C;2EAeA,yB;MAAA,4B;MAAA,oB;QAYkD,OAA0B,YAAZ,CAAY,C;O;KAZ5E,C;2EAeA,yB;MAAA,4B ;MAAA,oB;QAakD,OAA0B,YAAZ,CAAY,C;O;KAb5E,C;2EAeA,yB;MAAA,4B;MAAA,uB;QAS4D,OAAwC,Y AA1B,CAA0B,EA AZ,CAAY,C;O;KATpG,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAQiD,OAAA,MAA8B,MAAZ, CAAY,C;O;KAR/E,C;uEAUA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C; 2EAYA,yB;MAAA,4B;MAAA,oB;QackD,OAA0B,YAAZ,CAAY,C;O;KAd5E,C;uEA gBA,yB;MAAA,mC;MAA A,0B;Qac6D,OAAmC,IAA7B,CAA6B,EA AZ,IAAY,C;O;KadhG,C;qEA gBA,yB;MAAA,iB;MAAA,oB;QAW+C ,OAAA,MAA6B,KAAZ,CAAY,C;O;KAX5E,C;2EAaA,yB;MAAA,4B;MAAA,oB;QAOKD,OAA0B,YAAZ,CAA Y,C;O;KAP5E,C;yEASA,yB;MAAA,0B;MAAA,oB;QAOiD,OAAyB,WAAZ,CAAY,C;O;KAP1E,C;yEASA,yB; MAAA,4B;MAAA,oB;QAgBiD,OAA0B,YAAZ,CAAY,C;O;KAhB3E,C;yEAKBA,yB;MAAA,iB;MAAA,oB;QAU iD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAA+B,O AAZ,CAAY,C;O;KAVjF,C;iFAYA,yB;MA5hBA,4B;MA4hBA,oB;QAUqD,OA5hBE,YA4hBS,CA5hBT,C;O;KA khBvD,C;2EAYA,yB;MAAA,uC;MAAA,oB;QAQkD,OAAoB,MAAZ,CAAY,C;O;KARtE,C;uEAWA,yB;MAAA, iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;yEAYA,yB;MAAA,0B;MAAA,oB;QAWi D,OAAyB,WAAZ,CAAY,C;O;KAX1E,C;wEAeA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAA J,EA AO,CAAP,C;O;KAPrE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EA AO,C AAP,C;O;KAPrE,C;yEAUA,yB;MAAA,iB;MAAA,oB;QAaiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAb/E,C;sEA mBA,yB;MAAA,iB;MAAA,+B;QAasD,OAAA,MAA8C,KAA1B,SAA0B,EA AZ,CAAY,C;O;KAbpG,C;uEAeA,y B;MAAA,iB;MAAA,+B;QAOoD,OAAA,MAA8C,KAA1B,SAA0B,EA AZ,CAAY,C;O;KAPIG,C;kGAmBoD,yB; MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAGC,KAAZ,SAAY,C;O;KAAxC,C;gFAaT,yB;MAAA,0B;MAAA,4B;Q AAQ,OAA4B,WAAZ,SAAY,C;O;KAApC,C;gFAE3C,yB;MAAA,6C;MAAA,kC;QA08D,OAA0C,SAArC,SAaq C,EA AZ,IAAY,C;O;KAPxG,C;iFASA,yB;MAAA,6C;MAAA,kC;QAK4D,OAA0C,SAArC,SAaqC,EA AZ,IAAY, C;O;KALtG,C;oFAQA,yB;MAAA,iD;MAAA,4B;QAYmD,OA AW,WAA X,SAAW,C;O;KAZ9D,C;sFAcA,yB;M AAA,mD;MAAA,4B;QAYqD,OA AW,YAAX,SAAW,C;O;KAZhE,C;IAoBA,kB;MAUqC,OAAI,IAAI,CAAR,GA

AY,CAAC,CAAD,OAAM,CAAIB,GAA0B,C;K;wEAE/D,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KA  
AI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;wEAOA,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KA  
AI,CAAJ,  
EAAO,CAAP,C;O;KAL/D,C;mGAIbGd,yB;MAAA,mC;MAAA,4B;QAAQ,WAAI,SAAJ,C;O;KAAR,C;IAShB,+  
B;MAC5B,gBAAO,CAAP,C;QADoC,OACxB,E;WACZ,gBAAO,CAAP,C;QAFoC,OAExB,C;;QAFwB,OAG5B,C  
;K;IAKZ,kB;MASuC,OAAI,eAAI,CAAR,GAAY,CAAD,aAAX,GAAmB,C;K;wEAE1D,gB;MAKuD,OAAI,kBA  
AK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C;K;wEAE1E,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAA  
Z,GAAmB,C;K;mGAYxB,yB;MAAA,mC;MAAA,4B;QAAQ,WAAI,SAAJ,C;O;KAAR,C;IASjB,+B;MAC7B,2B  
AAO,CAAP,C;QADqC,OACzB,E;WACZ,2BAAO,CAAP,C;QAFqC,OAEB,C;;QAFyB,OAG7B,C;K;IC5mCZ,4  
B;MAI4C,qBAAQ,S;K;IAEpD,4B;MAI2C,qBAAQ,S;K;IAEnD,+B;MAGiD,qBAAQ,wCAAO,kBAAf,IAAoC,cA  
AQ,wCAAO,kB;K;IAEpG,iC;MAGgD,qBAAQ,uCAAM,kBAAd,IAAmC,cAAQ,uCAAM,kB;K;IAEjG,6B;MAG+  
C,QAAC,qBAAD,IAAiB,CAAC,kB;K;IAEjE,+B;MAG8C,QAAC,uBAAD,IAAiB,CAAC,kB;K;IAGhE,iC;MAOI,  
QAAQ,S;MACR,IAAI,CAAC,IAAM,UAAP,KAAsB,CAAE,KAAK,CAAP,GAAC,UAAP,C,K;MACJ,IAAI,CAAC,  
IAAM,SAAP,KAAsB,CAAE,KAAK,CAAP,GAAC,SAAP,C,K;MACJ,IAAI,CAAC,IAAM,SAAP,KAAsB,CAAE,K  
AAK,CAAP,GAAC,SAAP,C,K;MACJ,IAAI,CAAC,IAAM,QAAP,KAAsB,CAAE,KAAK,CAAP,GAAC,QAAP,C,K;  
MACJ,IAAI,CAAC,IAAM,KAAP,KAAsB,CAAE,KAAK,EAA7B,K;MACJ,OAAO,C;K;kGAGX,yB;MAAA,4B;  
MAAA,4B;QAM2D,mBAAy,SAAZ,C;O;KAN3D,C;IAQA,0C;MAOI,YATuD,YASvB,EAAf,aAAQ,CAAC,SAA  
D,IAAR,CAAe,CATuB,CASvD,I;K;IAEJ,sC;MAOI,OAAI,cAAQ,CAAZ,GAAe,CAAF,GAAsB,CAAE,IAAI,EAA  
J,GAIB+B,sB;K;IAoB3D,qC;MAQI,oBAAS,CAAC,SAAD,IAAT,C;K;IAEJ,yC;MAaI,oBAAI,QAAJ,GAaiB,cAA  
K,EAAL,GAAqB,Q;K;IAG1C,0C;MAaI,oBAAI,EAAJ,GAAoB,QAAPB,GAAiC,cAAK,Q;K;IAG1C,mC;MAMI,O  
AAK,arDhEmD,uBqDgEnD,CAAL,GAA0B,arDjE6B,sBqDiE7B,CAA1B,I;K;IAEJ,2C;MAMU,WAAW,SrDxEuC,  
c;MqDyEpD,e;QADJ,OACS,KA7E8C,YrDGA,sBqDHA,CA6E9C,I;;QADT,OA5EuD,YA8E3C,IA9E2C,C;;K;IAi  
F3D,4C;MAMU,UAAU,SrDpFuC,a;MqDqFnD,c;QADJ,OACS,KAAqB,sBrDpF0B,uBqDoF1B,CAArB,I;;QADT,  
OAEgB,sBAAJ,GAAL,C;K;IAGpB,wC;MAOU,WAAW,SrD/FuC,c;MqDgGpD,e;QAAK,UAAAS,kBrDjGqC,sBqDi  
GrC,C;QADIB,OrdjG4C,MAAa,KAAK,UAAAS,GAAT,EqDkGvB,CrDlGuB,C;;QqDmGID,aAAa,kBAAL,IAAK,C  
;QAFzB,OrdjG4C,MAAa,KAAK,UqDmG7C,CrDnG6C,EAAC,MAAd,C;;K;IqDsGIE,uC;MAOU,UAAU,SrD5Gu  
C,a;MqD6GnD,c;QAAK,WAAa,iBrD5GkC,uBqD4GIC,C;QADtB,Ord7G4C,MAAa,KAAK,UqD8GhD,CrD9GgD,  
EAAC,IAAd,C;;QqD+GID,YAAS,iBAAJ,GAAL,C;QAFrB,Ord7G4C,MAAa,KAAK,UAAAS,KAAT,EqD+GrB,CrD  
/GqB,C;;K;IqDkHIE,2C;MAaI,IAAI,CAAC,WAAa,EAAd,MAAqB,CAAzB,C;QACI,UAAU,SrD/HyC,a;QqDgInD  
,WAAW,SrD/HyC,c;QqDgIpD,aAAa,GAAL,IAAI,QAAR,GAAqB,IAAK,MAAK,CAAC,QAAD,IAAL,C;QACvC,  
cAAc,IAAK,IAAI,QAAT,GAAsB,GAAL,MAAK,CAAC,QAAD,IAAL,C;QACxC,OAAW,CAAC,WAAa,EAAd,M  
AAqB,CAAhC,GrDpIwC,MAAa,KAAK,UqDoIIB,MrDpIkB,EqDoIV,OrdpIU,CqDoIID,GrDpIwC,MAAa,KAAK,  
UqDoIS,OrdpIT,EqDoIkB,MrDpIIB,C;;QqDsInD,Q;QAAA,IAAI,CAAC,WAAa,EAAd,MAAqB,CAAzB,C;UAA  
A,OAA4B,S;;uBrDpLiB,uB;UqDoIP,arDrIM,sB;UqDqI5C,OrdtIiC,MAAa,KAAK,kBAAC,MAAd,C;;QqDsIID,W;;  
K;kFAKR,yB;MAAA,4C;MAAA,sC;QAaiE,6BAAW,CAAC,QAAD,IAAX,C;O;KAbjE,C;qECvKA,kC;MAII,OA  
AO,SAA8B,MAAK,WAAL,C;K;uEAGzC,8C;MAII,OAAO,SAA8B,MAAK,WAAL,EAAB,UAAIB,C;K;ICtCzC,  
iC;MACI,gBAAW,IAAI,OAAO,EAAG,GAEE,IAAI,IAAI,CAAC,CAAD,EAAL,EAJ,CAA5B,GAAuC,CAA9C,  
C;K;;IAKJ,sC;MACI,cAAO,QAAP,GAAB,QAAQ,Q;K;ICP9B,yC;K;;IAWA,+B;K;;4GAYA,yB;MAAA,gC;MA  
AA,yD;MAAA,sC;QAQI,OAAK,qBAAL,SAAK,iB;O;KART,C;ICPI,2B;MAAS,Q;MAAD,OAAwB,CAAvB,iEA  
AuB,Q;K;IAMhC,+B;MAAQ,iBAAU,SAAV,C;K;;;ICtB+B,4B;MACvC,8B;K;gEAAA,Y;MAAA,4B;K;2FAII,  
Y;MtG04B,MAAM,yB;K;kCsGLtC,iB;MACI,OAAO,oCAA0B,oBAAU,KAAM,OAAhB,C;K;oCAGrC,Y;MAC+  
B,gB;MAAA,8FAA0B,C;K;oCAEzD,Y;MAEI,OAAO,oBAAQ,eAAR,C;K;;IAIyB,kC;MAAuB,sBAAC,MAAd,C;  
MACL,Q;MAAtD,4BAAmC,CAAmB,OAAZ,MAAY,WAAAnB,kC;K;8FAAnC,Y;MAAA,gC;K;oDAEA,iB;MAC  
W,cAAgB,W;MAAvB,OpEoEuD,MAAa,QoEpEpD,KpEoEoD,EAAY,OAAZ,C;K;;IoEhEjC,0E;MAIvC,sBAAC,M  
AAd,C;MAFA,wC;MACA,8C;K;2CAEA,iB;MACI,IAAI,0CAAJ,C;QAAsC,OAAO,K;MAC7C,OAAa,uCAAO,K  
AAP,CAAN,IAAuB,+BAAmB,KAAM,kBAAZB,C;K;iGAGD,Y;MAAQ,6B;K;uDAEzC,iB;MACI,OAAO,0BAA  
mB,KAAAnB,C;K;;IAIf,6B;MAAA,iC;MAAoC,sBAAoB,MAAPB,C;MACHC,4BAAKC,S;K;+FAAIC,Y;MAAA,gC  
;K;qDAEA,iB;MAAgD,Y;K;2FAG5C,Y;MAAQ,MAAM,qCAA8B,6CAA9B,C;K;yCAEIB,iB;MAA4C,iBAAU,I;  
K;2CAEtD,Y;MAA+B,Q;K;;IAVnC,yC;MAAA,wC;QAAA,uB;;MAAA,iC;K;IAaA,uB;K;yFACqC,Y;MzG0EY,



MAAM,6ByG1EJ,oCzG0EkC,WAA9B,C;K;4FyGzEf,Y;MzGyES,MAAM,6ByGzED,uCzGyE+B,WAA9B,C;K;+  
CyGvEnD,iB;MzGuE6C,MAAM,6ByGvEG,uCzGuE2B,WAA9B,C;K;mCyGrEnD,iB;MAA4C,iBAAU,I;K;qCAEt  
D,Y;MAA+B,Q;K;;oHCnE/B,qB;MAAQ,2B;K;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,ICKZ,gE;MAMI,qBAAU,UAAV,EAAgC,OA  
AV,WAAU,CAAhC,EAA0C,gBAA1C,C;K;IAEJ,8B;MAC2C,iC;K;IAE3C,kC;MAC+C,qBAAU,cAAA,KAAM,W  
AAN,CAAV,EAA8B,KAAM,UAApC,EAA+C,IAA/C,C;K;IAE/C,2D;MAM0B,IAAN,I;MAAA,QAAM,QAAN,C;  
aACZ,I;;UAAA,K;aACA,K;;UAAA,K;;UAFY,K;;MAAhB,oB;MAMA,OAAO,uBAAMb,IAAnB,EAAqC,OAAZ,  
WAAY,CAArC,EAA+C,SAA/C,EAA0D,KAA1D,C;K;IAGX,kC;MAEI,OAAA,uCAAgB,K;K;IAEpB,8C;MAEI,O  
AAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,8C;MAEI,OAAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,kD;MAEI,O  
AAA,uCAAgB,uBAAc,IAAd,C;K;IC/CI,8D;MACpB,sC;MACA,sC;MACA,kD;K;mEAFa,Y;MAAA,gC;K;kEAC  
A,Y;MAAA,+B;K;yEACA,Y;MAAA,sC;K;iCAEA,iB;MACI,0CACQ,wBAAc,KAAM,WAApB,CADR,IAC0C,uB  
AAa,KAAM,UAAAnB,CAD1C,IAC0E,0BAAoB,KAAM,iB;K;mCAExG,Y;MACI,SAAC,CAAW,SAAX,eAAW,C  
AAX,GAAwB,EAAxB,QAAuC,SAAV,cAAU,CAAvC,IAAD,IAAsD,EAAtD,QAA4E,SAAjB,qBAAiB,CAA5E,I;  
K;mCAEJ,Y;MACkB,UACO,M;MADrB,aAAc,2D;MAEV,cAAU,IAAV,C;QAA6B,SAAX,eAAW,W;WAC7B,IA  
AA,MAAO,WAAP,S;QAAoC,SAAP,MAAO,W;;QAC5B,+B;MAHZ,2B;MAMA,WACQ,cAAU,UAAAd,GAAyB,E  
AAzB,GACe,eAAV,cAAU,EAAa,IAAb,EAAmB,GAAAnB,EAAwB,GAAxB,C;MACnB,eAAmB,qBAAJ,GAAsB,  
GAAtB,GAA+B,E;MAE9C,OAAO,iBAAiB,IAAjB,GAAwB,Q;K;;IAIvC,wB;MAAA,4B;MACI,4BAAwC,I;MAC  
xC,2BAAGD,W;MACHd,kCAAyC,K;K;0FAFzC,Y;MAAA,gC;K;yFACA,Y;MAAA,+B;K;gGACA,Y;MAAA,sC;  
K;sCACA,Y;MAAkC,gB;K;;IAJtC,oC;MAAA,mC;QAAA,kB;;MAAA,4B;K;IC7BsC,oE;MACIC,0B;MACA,wC;  
MACA,kC;MACA,oC;K;sEAHA,Y;MAAA,0B;K;6EACA,Y;MAAA,iC;K;0EACA,Y;MAAA,8B;K;2EACA,Y;M  
AAA,+B;K;4CAEA,Y;MAAkC,gB;K;;8CANtC,Y;MACI,gB;K;8CADJ,Y;MAEI,uB;K;8CAFJ,Y;MAGI,oB;K;8C  
AHJ,Y;MAII,qB;K;gDAJJ,kD;MAAA,8BACI,kCADJ,EAEI,uDAFJ,EAGI,8CAHJ,EAIL,iDAJJ,C;K;4CAAA,Y;M  
AAA,c;MACI,qD;MACA,4D;MACA,yD;MACA,0D;MAJJ,a;K;0CAAA,iB;MAAA,4IACI,oCADJ,IAEI,kDAFJ,IA  
GI,4CAHJ,IAII,8CAJJ,I;K;ICAA,4B;MAAA,gC;MAEI,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCA  
ApE,C;MAEf,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,EAA0E,mCAA1E,C;MAEIB,oBACmB,+B;MAEnB,  
oBACmB,wBAAoB,OAApB,EAAkE,SAAI,EAA6E,oCAA7E,C;MAEnB,iBACgB,wBAAoB,MAApB,EAA8D,M  
AA9D,EAASe,iCAAtE,C;MAEhB,kBACiB,wBAAoB,MAApB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MAEjB,gB  
ACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCAApE,C;MAEf,kBACiB,wBAAoB,MAApB,EAA+D,OAA/  
D,EAAwE,kCAAxE,C;MAEjB,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,EAA0E,mCAA1E,C;MAEIB,kBAC  
iB,wBAAoB,KAAPB,EAAiE,OAAjE,EAA0E,kCAA1E,C;MAEjB,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,  
EAA0E,mCAA1E,C;MAEIB,sBACqB,wBAAoB,KAAPB,EAAkE,WAAIE,EAA+E,sCAA/E,C;MAErB,yBACwB,  
wBAAoB,KAAPB,EAAqE,cAArE,EAAqF,yCAArF,C;MAExB,sBACqB,wBAAoB,WAApB,EAAwE,WAAxE,EA  
AqF,sCAArF,C;MAErB,sBACqB,wBAAoB,SAAPB,EAASe,WAAIE,EAAmF,sCAAnF,C;MAErB,uBACsB,wBA  
AoB,UAApB,EAAwE,YAAxE,EAAfF,uCAAtF,C;MAEtB,qBACoB,wBAAoB,UAApB,EAASe,UAAIE,EAAkF,q  
CAAI,C;MAEpB,sBACqB,wBAAoB,KAAPB,EAAkE,WAAIE,EAA+E,sCAA/E,C;MAErB,uBACsB,wBAAoB,Y  
AApB,EAA0E,YAA1E,EAAwF,uCAAXF,C;MAEtB,wBACuB,wBAAoB,YAApB,EAA2E,aAA3E,EAA0F,wCAA  
1F,C;K;IAMkB,qE;MAAA,qB;QAAE,01E/DD,00E+DU,EAAT,KAAiB,UAAjB,IAAkC,EAAY,OAAf,KAA0B,a;  
O;K;+CAJpG,iB;MAE2B,Q;MAAhB,U;MAAA,KAAGB,OAAhB,eAAGB,CAAI,KAJJ,CAAhB,U;QAAA,a;;QAC  
H,aAAa,wBAAoB,QAAPB,EAA+D,kBAA/D,EACoB,mDADpB,C;QAEG,eAAhB,UAAqC,M;QAHIC,SAIH,M;;  
MAJJ,a;K;IA7D+E,8C;MAAE,6B;K;IAGO,iD;MAAE,0B;K;IAME,kD;MAAE,8B;K;IAGZ,+C;MAAE,6B;K;IAG  
C,gD;MAAE,6B;K;IAGR,8C;MAAE,6B;K;IAGI,gD;MAAE,6B;K;IAGC,iD;MAAE,6B;K;IAGH,gD;MAAE,yB;K  
;IAGD,iD;MAAE,6B;K;IAGM,oD;MAAE,mC;K;IAGO,uD;MAAE,gC;K;IAGL,oD;MAAE,6B;K;IAGJ,oD;MAA  
E,6B;K;IAGE,qD;MAAE,8B;K;IAGR,mD;MAAE,4B;K;IAGJ,oD;MAAE,6B;K;IAGQ,qD;MAAE,8B;K;IAGC,sD;  
MAAE,+B;K;;IA5DvH,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;;ICCA,2B;MAEW,Q;MAAA,IAAI,KAA Y,SA  
Q,MAAR,CAAhB,C;QACH,kBAAW,MAAX,C;;QAEA,kBAAW,MAAX,C;;MAHJ,W;K;IAOJ,8B;MAC4E,QAA  
M,QAAS,OAAf,C;aACxE,C;UADwE,OACnE,WAAW,SAAS,CAAT,CAAX,C;aACL,C;UAFwE,OAEnE,+B;;UA  
FmE,OAGhE,iB;;K;IAGZ,oC;MAEU,IAAN,I;MAAA,Q3EhB0C,O2EgB3B,CAAf,C;aACI,Q;UAA6B,OAAjB,8B  
AAiB,Y;UAA7B,K;aACA,Q;UAA Y,OAAI,CAAY,CIEhC,GkEamC,CAAf,MAAkC,CAAtC,GAAyC,8BAAiB,S  
AA1D,GAAwE,8BAAiB,Y;UAArG,K;aACA,S;UAA8B,OAAjB,8BAAiB,a;UAA9B,K;aACA,U;UAA+B,OAAjB,

8BAAiB,eAAgB,CAAY,OAA5B,C;UAA/B,K;;UAGQ,6B;YAA5C,OAAjB,8BAAiB,kB;eACtC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,yB;YAAkC,OAAjB,8BAAiB,c;eAClC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,4B;YAAqC,OAAjB,8BAAiB,iB;eACrC,6B;;eACA,sB;YAAkC,OAAjB,8BAAiB,W;;YAE9B,kBAAkB,MAAA,gBAAe,CAAf,CAAkB,Y;YAE7C,oBAAgB,MAAhB,C;:cAAiD,OAAjB,8BAAiB,S;iBACjD,oBAAgB,KAAhB,C;cAAgD,OAAjB,8BAAiB,e;;cAE5C,cAA0B,W;cAC1B,kBAAW,OAAAX,C;;;UAxBxB,K;;MAAA,W;K;IAgCJ,4B;MAMW,Q;MAJP,IAAI,WAAW,MAAf,C;QAA6B,OAAO,8BAAiB,Y;;MAErD,eAA5B,MAAY,W;MAE3B,IAAI,gBAAJ,C;QACH,IAAI,QAAS,SAAT,QA AJ,C;UACl,aAAa,qBAAiB,MAAjB,C;UACb,oBAAsB,M;UACtB,a;;UAES,OAAT,QAAS,S;;;QAGb,4BAAiB,MAAjB,C;;MATJ,W;K;ICrCJ,0B;MAII,sBAAY,C;K;qEAchB,4B;MAIkE,iBAAY,KA AZ,C;K;2EAEIE,qB;MAI8D,gB;K;ICIDb,2C;MAC7C,qBAAwC,Q;K;iDAExC,Y;MACkC,IAAf,I;MAAA,yB;MA AA,iB;QAAe,MAAM,6BAAsB,0CAAiB,C;;MAApC,eAAe,I;MACf,qBAAc,I;MACd,OAAO,QAAS,W;K;;;ICLa,kD;MADrC,e;MACsC,0B;MAAyB,gB;MAD/D,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAEI,qEAGW,CAHX,EAGc,IAHd,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,+EAGgB,CAHhB,EAGmB,IAHnB,C;MAKA,yEAGa,CAHb,EAGgB,IAHh B,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,6EAGe,CAHf,EAGkB,IAHIB,C;MAKA,6FAGuB,CAHvB ,EAG0B,IAH1B,C;MAKA,yFAGqB,CAHrB,EAGwB,IAHxB,C;MAKA,4EAGc,EAHd,EAGkB,IAHIB,C;MAKA, 0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,8EAGe,EAHf,EAGmB,IAHn B,C;MAKA,wFAGoB,EAHpB,EAGwB,IAHxB,C;MAKA,gEAGQ,EAHR,EAGY,IAHZ,C;MAKA,8DAGO,EAHP ,EAGW,IAHX,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,oEAGU,EAHV,EAGc,IAHd,C;MAKA,kFAG iB,EAHjB,EAGqB,IAHrB,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB ,C;MAKA,4FAGsB,EAHtB,EAG0B,IAH1B,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,wEAGY,EAHZ ,EAGgB,IAHhB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MA KA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,oGAG0B,EAH1B,EAG8B,IAH9B,C;MAKA,gGAGwB,EAHxB,EAG 4B,IAH5B,C;MAUA,oC;K;;IA3JA,+C;MAAA,yB;MAAA,uC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD; MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MA AA,yB;MAAA,yC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,2D;MAA A,yB;MAAA,mD;K;;IAKA,yD;MAAA,yB;MAAA,iD;K;;IAKA,kD;MAAA,yB;MAAA,0C;K;;IAKA,iD;MAAA,y B;MAAA,yC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,wD;MAAA,yB; MAAA,gD;K;;IAKA,4C;MAAA,yB;MAAA,oC;K;;IAKA,2C;MAAA,yB;MAAA,mC;K;;IAKA,gD;MAAA,yB;M AAA,wC;K;;IAKA,8C;MAAA,yB;MAAA,sC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,sD;MAAA,yB;MAA A,8C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,0D;MAAA,yB;MAAA,kD;K;;IAKA,sD;MAAA,yB;MAAA,8 C;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,oD;MAAA,yB;MAAA,4C; K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,8D;MAAA,yB;MAAA,sD;K;;IAKA,4D;MAAA,yB;MAAA,oD;K;8 CAKA,gB;MAG2D,OAAK,iBAAL,IAAK,CAAL,KAA2B,IAAK,c;K;IAE3F,kC;MAAA,sC;K;uDACI,oB;MAEQ,I ADE,QACF,IAAG,CAAH,IADE,QACF,IAAM,EAAN,C;QADJ,OACgB,sBAAS,QAAT,C;WACZ,IAFE,QAEF,IA AG,EA AH,IAFE,QAEF,IAAO,EAAP,C;QAFJ,OAEiB,sBAAS,WAAW,CAAX,IAAT,C;;QACL,MAAM,gCAAyB ,eAAY,QA AZ,qBAAzB,C;K;;IAL1B,8C;MAAA,yB;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7JJ,+B;MAAA,+yC ;K;;IAAA,oC;MAAA,a;aAAA,Y;UAAA,4C;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aA AA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,kB;UAAA,kD;aAAA,gB;UAAA,gD;aAAA,wB;UAAA,wD;aAAA,sB ;UAAA,sD;aAAA,e;UAAA,+C;aAAA,c;UAAA,8C;aAAA,iB;UAAA,iD;aAAA,gB;UAAA,gD;aAAA,qB;UAAA,q D;aAAA,S;UAAA,yC;aAAA,Q;UAAA,wC;aAAA,a;UAAA,6C;aAAA,W;UAAA,2C;aAAA,kB;UAAA,kD;aAAA, mB;UAAA,mD;aAAA,iB;UAAA,iD;aAAA,uB;UAAA,uD;aAAA,mB;UAAA,mD;aAAA,a;UAAA,6C;aAAA,iB;U AAA,iD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,2B;UAAA,2D;aAAA,yB;UAAA,yD;;UAAA,6D;;K;;ICKi D,2C;uBAA+B,O;;K;;IAC5E,8C;MAAA,kE;MAAuB,qCAAK,IAAL,C;MAAvB,Y;K;ICD8B,gC;MAE9B,gBAAiC ,YAA Y,SAAhB,GAA2B,OAA3B,GAAwC,E;K;uFAGjE,Y;MAAQ,OAAO,aAAY,O;K;yCAE/B,iB;MACW,gBAA P,a;MIGqGG,Q;MAAA,IkGrGc,KIGqGV,IAAS,CAAT,IkGrGU,KIGqGL,IAAS,2BAA3B,C;QAAA,OAA5C,qBkGr GxB,KIGqGwB,C;;QkGrGf,MAAM,8BAA0B,mCAAyB,WAAzB,MAA1B,C;;MAAhC,W;K;kDAEJ,gC;MAAgF, OAAA,avG0NY,WuG1NK,UvG0NL,EUg1NiB,QvG0NjB,C;K;6CuGxN5F,iB;MACl,qCAAU,KAAV,C;MACA,O

AAO,I;K;6CAGX,iB;MACI,iBAAgB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,uC;MACI,OAAA,IAAK,qB  
AAAY,wBAAS,MAArB,EAA6B,UAA7B,EAAYC,QAAzC,C;K;sCAET,Y;MAayB,UAEK,M;MAL1B,eAAe,E;MA  
Cf,YAAY,aAAO,OAAP,GAAgB,CAAhB,I;MACZ,OAAO,SAAS,CAAhB,C;QACI,UAAU,0BAAO,YAAP,EAAO  
,oBAAP,Q;QACV,IAAQ,eAAJ,GAAl,CAAJ,IAAwB,SAAS,CAArC,C;UACI,WAAW,0BAAO,cAAP,EAAO,sBA  
AP,U;UACX,IAAS,gBAAL,IAAK,CAAT,C;YACI,WAAW,+BAAW,iBAAX,wBAaKB,gBAAIB,C;;YAEX,WAA  
W,+BAAW,gBAAX,wBAAiB,iBAAjB,C;;;UAGf,gCAAY,GAAZ,C;;;MAGR,gBAAS,Q;MACT,OAAO,I;K;6CAG  
X,iB;MAOI,iBAAgB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAQI,iBAAU,K;MACV,OAAO,I;K;6CA  
GX,iB;MAQI,iBAAgB,eAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAOI,gBAAA,IAAK,SAAL,IAAe,wBA  
AS,MAAxB,C;MACA,OAAO,I;K;uCAGX,Y;MAU6B,kB;K;qDAE7B,2B;K;8CAcA,kB;MAO0C,OAAA,IAAY,S  
AAAY,SAAQ,MAAR,C;K;8CAEIE,8B;MAQ2D,OAAA,IAAY,SAAY,SAAQ,MAAR,EAAGB,UAAhB,C;K;kDAEn  
F,kB;MAQ8C,OAAA,IAAY,SAAY,aAAY,MAAZ,C;K;kDAEtE,8B;MASI,IAAI,MjG0GwC,YAAU,CiG1GID,IA  
AoB,aAAa,CAArC,C;QAAwC,OAAO,E;MAC/C,OAAO,IAAY,SAAY,aAAY,MAAZ,EAAoB,UAApB,C;K;4CA  
GnC,wB;MAWI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avG+C+E,WuG/C9D,CvG+C8D,  
EuG/C3D,KvG+C2D,CuG/C/E,YAA6B,KAA7B,IAAqC,avG4C2B,WuG5CV,KvG4CU,C;MuG3CzE,OAAO,I;K;6  
CAGX,wB;MAQI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGiC+E,WuGjC9D,CvGiC8D,  
EuGjC3D,KvGiC2D,CuGjC/E,uBAA6B,kBAA7B,IAAqC,avG8B2B,WuG9BV,KvG8BU,C;MuG7BzE,OAAO,I;K;  
6CAGX,wB;MAUI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGiB+E,WuGjB9D,CvGiB8D  
,EuGjB3D,KvGiB2D,CuGjB/E,GAAmC,eAAN,KAAM,CAAnC,GAAsD,avGcU,WuGdO,KvGcP,C;MuGbzE,OA  
AO,I;K;6CAGX,wB;MAAI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGF+E,WuGE9D,CvG  
F8D,EuGE3D,KvGF2D,CuGE/E,GAAmC,SAAN,KAAM,CAAnC,GAAgD,avGLgB,WuGKC,KvGLD,C;MuGMz  
E,OAAO,I;K;6CAGX,wB;MAWI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGnB+E,WuG  
mB9D,CvGnB8D,EuGmB3D,KvGnB2D,CuGmB/E,GAAmC,SAAN,KAAM,CAAnC,GAAgD,avGtBgB,WuGsBC,  
KvGtBD,C;MuGuBzE,OAAO,I;K;6CAGX,wB;MAUI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,eAA  
e,wBAAS,M;MACxB,gBAAc,IAAK,SvGpCqE,WuGoCpD,CvGpCoD,EuGoCjD,KvGpCiD,CuGoC1E,GAAkC,Q  
AAIC,GAA6C,IAAK,SvGvCS,WuGuCQ,KvGvCR,C;MuGwCzE,OAAO,I;K;gDAGX,qB;MAcI,IAAI,YAAY,CA  
AhB,C;QACI,MAAM,gCAAYB,0BAAuB,SAAvB,MAAZB,C;;MAGV,IAAI,aAAa,WAAjB,C;QACI,gBAAS,avG3  
D2E,WuG2D1D,CvG3D0D,EuG2DvD,SvG3DuD,C;;QuG6DpF,aAAU,WAAV,MAAuB,SAAvB,M;UACI,qCAA  
U,CAAV,C;;K;gDAKZ,sB;MAQI,oCAAA,4BAAmB,UAAAnB,EAA+B,WAA/B,C;MAEb,OAAO,avGhFkE,WuGg  
FjD,UvGhFiD,C;K;gDuGmF7E,gC;MAQI,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAYC,WAAzC,C;MAEb,  
OAAO,avG1FiF,WuG0FhE,UvG1FgE,EuG0FpD,QvG1FoD,C;K;yCuG6F5F,Y;K;uCACa,Y;MAAKC,oB;K;oCAEI  
C,Y;MAOI,gBAAS,E;MACT,OAAO,I;K;0CAGX,wB;MAQI,oCAAA,2BAaKB,KAAIB,EAAYB,WAAzB,C;MAEb  
,gBAAS,avGII+E,WuGki9D,CvGII8D,EuGki3D,KvGII2D,CuGki/E,uBAA6B,kBAA7B,IAAqC,avGrI2B,WuGqIV  
,QAAQ,CAAR,IvGrIU,C;K;+CuGwI7E,uC;MAYI,yBAaKB,UAAIB,EAA8B,QAA9B,EAAwC,WAAxC,C;MAEA  
,gBAAc,IAAK,SvGnJqE,WuGmJpD,CvGnJoD,EuGmJjD,UvGnJiD,CuGmJ1E,GAAuC,KAAvC,GAA+C,IAAK,Sv  
GtJO,WuGsJU,QvGtJV,C;MuGuJzE,OAAO,I;K;kDAGX,wC;MACI,IAAI,aAAa,CAAb,IAAKB,aAAa,MAAnC,C;  
QACI,MAAM,8BAA0B,iBAAc,UAAAd,kBAAmC,MAA7D,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAA  
yB,gBAAa,UAAb,qBAAqC,QAArC,MAAZB,C;;K;+CAId,iB;MAYI,oCAAA,2BAaKB,KAAIB,EAAYB,WAAzB,C  
;MAEb,gBAAS,avG9K+E,WuG8K9D,CvG9K8D,EuG8K3D,KvG9K2D,CuG8K/E,GAA6B,avGjLmC,WuGiLIB,Q  
AAQ,CAAR,IvGjLkC,C;MuGkLzE,OAAO,I;K;kDAGX,gC;MAWI,yBAaKB,UAAIB,EAA8B,QAA9B,EAAwC,  
WAAxC,C;MAEA,gBAAS,avG/L+E,WuG+L9D,CvG/L8D,EuG+L3D,UvG/L2D,CuG+L/E,GAAkC,avGIM8B,Wu  
GkMb,QvGIMa,C;MuGmMzE,OAAO,I;K;kDAGX,gE;MAc+C,iC;QAAA,oBAAyB,C;MAAG,0B;QAAA,aAAkB,  
C;MAAG,wB;QAAA,WAAgB,IAAK,O;MAKIF,IACf,I;MALhB,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAY  
C,WAAzC,C;MACb,oCAAA,4BAAmB,iBAAnB,EAAc,oBAAoB,QAApB,GAA+B,UAA/B,IAAtC,EAIf,WA  
AY,OAA7F,C;MAEb,eAAe,iB;MACf,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,YAAY,eAAZ,EAAY,uBAAZ,UA  
A0B,yBAAO,KAAP,C;;K;kDAIIC,uC;MAcI,iBAAgB,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,C;MACHB,  
OAAO,I;K;kDAGX,uC;MAYI,gBAAgB,KAAM,W;MACTB,oCAAA,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAYC,  
SAAU,OAAnD,C;MAEb,iBAAU,SvG5P8E,WuG4P1D,UvG5P0D,EuG4P9C,QvG5P8C,C;MuG6PxP,OAAO,I;K;k  
DAGX,8C;MagBI,oCAAA,4BAAmB,KAAAnB,EAA0B,IAAK,OAA/B,C;MAEb,gBAAS,avGIR+E,WuGkr9D,CvG

IR8D,EuGkR3D,KvGIR2D,CuGkR/E,GAAMC,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,CAAnC,GAA0E,av  
GrRV,WuGqR2B,KvGrR3B,C;MuGsRzE,OAAO,I;K;kDAGX,8C;MAGBI,oCAAA,4BAAMb,KAAnB,EAA0B,WA  
A1B,C;MAEb,gBAAgB,KAAM,W;MACtB,oCAAA,4BAAMb,UAAAnB,EAA+B,QAA/B,EAAyC,SAAU,OAAAnD,  
C;MAEb,gBAAS,avG3S+E,WuG2S9D,CvG3S8D,EuG2S3D,KvG3S2D,CuG2S/E,GAA6B,SvG3SkD,WuG2S9B,U  
vG3S8B,EuG2SIB,QvG3SkB,CuG2S/E,GAAYe,avG9ST,WuG8S0B,KvG9S1B,C;MuG+SzE,OAAO,I;K;;IA5hBX,  
6C;MAAA,uD;MAKoC,2B;MALpC,Y;K;IAQA,8C;MAAA,uD;MAC4C,0BAAK,OAAQ,WAAb,C;MAD5C,Y;K;I  
AGA,qC;MAAA,uD;MACuB,0BAAK,EAAL,C;MADvB,Y;K;2EAshBJ,qB;MAOgE,OAAA,SAAK,Q;K;uEAERe,  
mC;MAQ+E,SAAK,aAAI,KAAJ,EAAW,KAAX,C;K;+EAEPf,kD;MAaI,OAAA,SAAK,kBAAS,UAAAT,EAAqB,Q  
AArB,EAA+B,KAA/B,C;K;+EAET,4B;MAY6E,OAAA,SAAK,kBAAS,KAAT,C;K;qFAEIF,2C;MAWoG,OAAA,  
SAAK,qBAAY,UAAZ,EAAb,W,QAAxB,C;K;uFAEzG,2E;MAe2E,iC;QAAA,oBAAYB,C;MAAG,0B;QAAA,aAA  
kB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC7I,SAAK,qBAAY,WAAZ,EAAyB,iBAAzB,EAA4C,UAA5C,E  
AAwD,QAAxD,C;K;qFAET,kD;MAeI,OAAA,SAAK,qBAAY,KAAZ,EAAMb,UAAAnB,EAA+B,QAA/B,C;K;uFA  
ET,kD;MAaI,OAAA,SAAK,qBAAY,KAAZ,EAAMb,UAAAnB,EAA+B,QAA/B,C;K;qFAET,yD;MAiBI,OAAA,S  
AAK,qBAAY,KAAZ,EAAMb,KAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;uFAET,yD;MAiBI,OAAA,SAAK,q  
BAAY,KAAZ,EAAMb,KAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;qFxG1rBT,qB;MAMoD,OA6BW,8BAAY,  
cAfrB,YAAY,CAAZ,C;K;yFAZtD,qB;MAYsD,OAeS,8BAAY,cAfrB,YAAY,CAAZ,C;K;iFAEtD,qB;MAaoD,OA  
AW,8BAAY,c;K;qFAE3E,yB;MAAA,uD;MAAA,4B;QAMoD,+B;O;KANpD,C;IAQA,kC;MAYI,gBAiB2D,8BA  
AY,c;MAhBvE,OAAW,SAAU,OAAV,GAAMb,CAAvB,GAA0B,SAA1B,GAAoC,qBAAU,CAAV,C;K;iFAG/C,q  
B;MAaoD,OAAW,8BAAY,c;K;IAE3E,kC;MAU+C,mC;K;IAE/C,oC;MAGoD,QAAQ,cAAA,sCAAK,mBAAL,E  
AAyB,sCAAK,mBAA9B,CAAR,6B;K;IAEpD,mC;MAGmD,QAAQ,cAAA,sCAAK,kBAAL,EAAwB,sCAAK,kB  
AA7B,CAAR,6B;K;IAO/C,iC;MAAQ,OAAA,oCAAA,iBAAQ,2BAAR,C;K;IAEzB,8B;MAOI,IAAI,YAAO,GAA  
X,C;QACI,OAAO,I;MAEX,OAAO,gCAA8C,mD;K;IAGzD,6B;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCA  
AoB,CAAQ,kBAAK,EAAL,CAAR,6BAAXB,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;  
MAEX,OAAO,uB;K;IAGX,oC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ,kBAAK,EAAL,CAA  
R,6BAApB,IAAwC,CAAQ,kBAAK,EAAL,CAAR,6BAA5C,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;Q  
ACI,OAAO,K;;MAGX,OAAO,0BAAiB,uB;K;IAG5B,4B;MASI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;Q  
ACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAAO,sB;K;IAGX,gC;MAUI,IAAI,CAA  
Q,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAA  
O,0B;K;IAGX,gC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GA  
AX,C;QACI,OAAO,K;;MAEX,OAAO,0B;K;IAGX,gC;MASI,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OA  
AO,gCAAoD,yD;K;IAG/D,iC;MAUI,OAAO,aAAQ,EAAR,IAAoB,CAAQ,mBAAU,GAAV,CAAR,6B;K;IAG/B,i  
C;MAMiD,kC;K;iFyGtPjD,yB;MAAA,+C;MAAA,4B;QAMuD,OAAK,UAAAL,SAAK,C;O;KAN5D,C;IAQA,gC;  
MAMiD,4B;MAAA,S;QAAGb,cAAA,SxG4LC,cwG5LD,EAAoB,MAApB,C;;MAAhB,W;K;IAEjD,6B;MAI0C,Q;  
MAAA,yDAaKB,kBAaKB,SAaIB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SAaIB,  
C;K;IAG3E,8B;MAI4C,Q;MAAA,0DAAMb,kBAaKB,SAaIB,C;K;IAE/D,qC;MAKsD,Q;MAAA,0CAAc,KAAd,  
oBAAwB,kBAaKB,SAaIB,C;K;IAE9E,0B;MAIwC,Q;MAAA,wDAAiB,kBAaKB,SAaIB,C;K;IAEzD,mC;MAKk  
D,Q;MAAA,wCAAY,KAAZ,oBAAsB,kBAaKB,SAaIB,C;K;IAExE,2B;MAI0C,Q;MAAA,yDAaKB,kBAaKB,SA  
aIB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SAaIB,C;K;IAE3E,6B;MAIyF,kBAA  
1C,CAAOS;MACID,IAAO,QIHED,WkHfC,CAAH,IAAc,CAAM,kBAApB,KIHED,WkHf6B,KAAM,GAAN,IAAk  
B,kBAajD,CAAJ,C;QACI,4B;MAFsC,OIHIBnC,W;K;6EkHZX,yB;MAAA,6C;MAAA,4B;QAKmD,0B;O;KALn  
D,C;IAOA,mC;MAIgG,kBAA1C,CAAOS;MAAR,OACjD,EAAK,QIH2BgB,WkH3BhB,CAAH,IAAc,CAAM,kB  
AApB,KIH2BmB,WkH3BY,KAAM,GAAN,IAAkB,kBAajD,CAAF,CIH2BO,GAAqB,WAArB,GAA+B,I;K;yFkH  
xB1C,yB;MAAA,yD;MAAA,4B;QAK0D,gC;O;KAL1D,C;iFAOA,yB;MAAA,6C;MAAA,mC;QAO6D,OAAa,SA  
AR,SAAQ,EAAS,KAAT,C;O;KAP1E,C;iFASA,yB;MAAA,6C;MAAA,mC;QAO8D,OAAa,SAAR,SAAQ,EAAS,  
KAAT,C;O;KAP3E,C;IASA,sC;MAMqD,OAAA,SAAY,UAAAS,WAAW,KAAX,CAAT,C;K;IAEjE,4B;MAAsC,Q  
AAM,SxG4EsB,cwG5E5B,C;aACIC,K;aAAA,M;aAAA,M;UADkC,OACT,I;UADS,OAE1B,K;K;IAGZ,2B;MA  
KI,IAAI,EAAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYB,WAAQ,KAAR,kCAAzB,C;;MAEV,OAA  
O,K;K;IAGX,8B;MAA2D,Q;MACvD,YAAQ,EAAR,IAAE,QAAQ,EAAvB,C;QAA8B,cAAO,E;WACrC,YAAQ,E

AAR,IAAe,QAAQ,EAAb,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,YAAQ,EAAR,IAAe,QAAQ,GAAvB  
,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,WAAO,GAAP,C;QAAMb,S;WACnB,YAAQ,KAAR,IAAoB,Q  
AAQ,KAA5B,C;QAAwC,cAAO,KAAP,GAakB,EAaIB,I;WACxC,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,C;QA  
AwC,cAAO,KAAP,GAakB,EAaIB,I;;QAC3B,sBAAL,IAAK,C;MIH9CN,a;MkHuCgD,OAQ/C,WAAJ,GAAiB,E  
AAjB,GAAyB,E;K;ICIJG,2C;MAHpC,e;MAGqC,kB;MAHrC,iB;MAAA,uB;K;IAAA,kC;MAAA,qC;O;MAII,qEA  
CY,GADZ,C;MAEA,iEAIU,GAJV,C;K;;IAFA,+C;MAAA,wB;MAAA,uC;K;;IAEA,6C;MAAA,wB;MAAA,qC;K;  
;IANJ,8B;MAAA,mF;K;;IAAA,mC;MAAA,a;AAA,a;UAAA,4C;aAAA,W;UAAA,0C;;UAAA,4D;;K;;IAawG,4B;  
MAAE,OAAA,EAAG,M;K;IAA7G,qC;MAAqE,iCAAA,EAAb,EAA0B,OAA1B,0BAAMC,cAAAnC,C;K;IAQIC,2B  
;MAAC,kB;K;;sCALpC,Y;MAKoC,iB;K;wCALpC,iB;MAAA,sBAKoC,qCALpC,C;K;oCAAA,Y;MAAA,OAKoC  
,iDALpC,M;K;oCAAA,Y;MAAA,c;MAKoC,sD;MALpC,a;K;kCAAA,iB;MAAA,2IAKoC,sCALpC,G;K;IAQA,g  
C;MAUsB,gB;MAAA,iF;MAAA,mB;QACX,MAAM,qCAA8B,8DAA9B,C;;MAdB,kBAaKB,M;MAGIB,OAAO,  
wBAAy,IAAZ,C;K;IAiBe,iC;MA4PtB,6B;MAnPA,eACoC,O;MACpC,eACsD,QAAR,OAAQ,C;MACtD,uBAAo  
C,WAAO,OAAP,EAawB,QAAR,OAAQ,EAAQ,IAAR,CAAxB,C;MACpC,6BAA2C,I;MAI3C,oCAAKD,I;K;OCA  
HID,Y;MACI,Q;MAAA,U;MAAA,gD;QAAA,a;;QAA8D,gBAAvC,WAAO,YAAP,EAawB,QAAR,YAAQ,EAAQ  
,IAAR,CAAxB,C;QAA8C,6BnHkBnE,S;QmHIBF,SnHmBG,S;;MmHnBH,a;K;iDAGJ,Y;MACI,Q;MAAA,U;MA  
AA,uD;QAAA,a;;QnH3BG,gB;QmH4BC,IAAY,aAAR,YAAQ,EAawB,EAAX,CAAR,IAAMC,WAAR,YAAQ,EA  
AS,EAAT,CAAvC,C;UAAA,eACI,oB;;UAEA,OAAO,WAAO,MAA2B,UAAf,YAAR,YAAQ,qBAAU,EAav,EA  
Ae,qBAAQ,EAAR,EAA3B,MAAP,EAA2D,QAAR,YAAQ,EAAQ,IAAR,CAA3D,C;QACb,4B;QAAO,oCnHSP,S;  
QmHdF,SnHeG,S;;MmHfH,a;K;sCAQJ,iB;MAEkB,MAAd,oBAAC,C;MACd,YAAy,oBAAC,MAAK,KAAM,WA  
AX,C;MAC1B,OAAO,iBAAiB,KAAM,MAAN,KAAe,CAAhC,IAAQ,C,oBAAC,UAAAd,KAA2B,KAAM,O;K;8CA  
GjF,iB;MAEkB,MAAd,oBAAC,C;MACd,OAAO,oBAAC,MAAK,KAAM,WAAX,C;K;wCAGzB,wB;MAGI,IAAI,  
QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OA  
A9E,C;;MAEV,cAAc,0B;MACd,oBAAoB,K;MACpB,OAAO,OAAQ,MAAK,KAAM,WAAX,C;K;mCAGnB,6B;  
MAS4C,0B;QAAA,aAaKB,C;MAC1D,IAAI,aAAa,CAAb,IAAKB,aAAa,KAAM,OAAzC,C;QACI,MAAM,8BAA  
0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAqB,SAAd,oBAAC,EAAS,KAAM,WAAf,EAA2B,  
UAA3B,EAAuC,oBAAvC,C;K;IAeG,6E;MAAA,mB;QAAE,+BAAK,aAAL,EAAy,kBAAZ,C;O;K;IAA2B,uC;M  
AAW,OAAA,KAAM,O;K;sCAZ1E,6B;MAQ+C,0B;QAAA,aAaKB,C;MAC7D,IAAI,aAAa,CAAb,IAAKB,aAAa,  
KAAM,OAAzC,C;QACI,MAAM,8BAA0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAO,mBA  
AiB,6CAAjB,EAA8C,sBAA9C,C;K;0CAGX,iB;MAMI,OAA2B,SAA3B,iCAA2B,EAAS,KAAM,WAAf,EAA2B,  
CAA3B,EAA8B,oBAA9B,C;K;sCAE/B,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,M  
AAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,OAA2B,SAAPB,0BAAoB,EAAS,KAAM  
,WAAf,EAA2B,KAA3B,EAaK,C,oBAAIC,C;K;IA2BL,mD;MAAA,qB;QAAE,2BAAoB,EAAPB,EAawB,mBAAx  
B,C;O;K;sCAvB5B,8B;MAoBI,IAAI,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAb,IAA+B,CAAA,YAAZ,WAAy,E  
AAS,EAAT,CAAhD,C;QACI,OAAO,KAAM,WzGoF4E,SyGpFnD,oBzGoFmD,EyGpFpC,WzGoFoC,C;;MyGIF7  
F,OAAO,qBAAQ,KAAR,EAae,iCAaf,C;K;sCAGX,4B;MAMI,YAAy,kBAAK,KAAL,C;MACZ,IAAI,aAAJ,C;Q  
AAmB,OAAO,KAAM,W;MAEhC,gBAAGB,C;MACHB,aAAa,KAAM,O;MACnB,SAAS,mBAAc,MAAd,C;;QAE  
L,iBAAiB,oB;QACjB,EAAG,gBAAO,KAAP,EAac,SAAd,EAAYB,UAAW,MAAM,MAA1C,C;QACH,EAAG,gB  
AAO,UAAU,UAAV,CAAP,C;QACH,YAAy,UAAW,MAAM,aAAjB,GAAgC,CAAhC,I;QACZ,QAAQ,UAAW,O  
;;MACd,oBAAy,MAAZ,IAAsB,aAAtB,C;MAET,IAAI,YAAy,MAAhB,C;QACI,EAAG,gBAAO,KAAP,EAac,S  
AAd,EAAYB,MAAZB,C;;MAGP,OAAO,EAAG,W;K;2CAGd,8B;MAyB+B,IAAf,I;MALZ,IAAI,CAAA,YAAZ,W  
AAy,EAAS,EAAT,CAAb,IAA+B,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAhD,C;QACI,uBAA+B,QAAR,YAAQ  
,EAAQ,GAAR,C;QAC/B,OAAO,KAAM,WzG8B4E,SyG9BnD,WAAO,YAAP,EAAGB,gBAAhB,CzG8BmD,EyG  
9BhB,WzG8BgB,C;;MyG3BjF,yBAAK,KAAL,C;MAAA,iB;QAAe,OAAO,KAAM,W;;MAAxC,YAAy,I;MCqK0  
,gBAAhB,sB;MDIKC,yBnG4KGF,0BmG5KzD,CnG4KyD,EmG5KhD,WAAM,MnG4K0C,CAAKC,WmG5KIH,C;  
MACA,yBAAO,uCAAP,C;MACA,yBnG0KgF,0BmG1KnD,WAAM,KAAZ,GAAmB,CAAnB,InG0KyD,EmG1K7  
B,YnG0K6B,CAAKC,WmG1KIH,C;MAHJ,OnHjKG,SoHoUqC,W;K;oCD5J5C,wB;MAO6C,qB;QAAA,QAAa,C;  
MAMxC,Q;MALd,wBAAwB,KAAxB,C;MnHpJG,SmHqJW,qBAAQ,KAAR,C;MAAd,cAAuC,UAAS,CAAb,GA  
AgB,EAAhB,GAA2B,OAAH,EAAG,EAAK,QAAQ,CAAR,IAAL,C;MAC9D,ahI1KgD,gB;MgI2KhD,gBAAGB,C;

MAEF,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAA  
M,MAAM,MAAnC,CAA0C,WAApD,C;QACP,YAAY,KAAM,MAAM,aAAZ,GAA2B,CAA3B,I;;MAEhB,MAA  
O,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,OAA7B,CAAqC,WAA/C,C;MACP,OAAO,M;K;IAgB  
S,yI;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,0C;MAAA,oC;MAAA,0C;MAAA,yB;MAAA,6B;MAAA,8B;MA  
AA,8B;MAAA,kC;K;;;gEAAA,Y;;;iCACA,mCAAK,wBAAL,C;cACZ,IAAI,4BAAiB,6BAAS,CAA9B,C;gBAC  
I,gB;gCAAA,iCAAM,wBAAM,WAAZ,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,gB;;;;;cAEI,M;;qCAGY,C  
;sCACC,C;cAEjB,gB;;;sCACqB,+B;cACjB,gB;8BAAA,iCnGwH4E,mBmGxHtE,wBnGwHsE,EmGxHtD,oBnGw  
HsD,EmGxH3C,qBAAW,MAAM,MnGwH0B,CAAkC,WmGxH9G,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,u  
BAAy,qBAAW,MAAM,aAAjB,GAAgC,CAAhC,I;cACZ,mBAAQ,qBAAW,O;cAJvB,KAKS,qDALt,EAKS,qBA  
LT,OAKyB,2BAAQ,CAAR,IALzB,KAKsC,gBALtC,S;gBAAA,gB;;;cAAA,gB;;cAOA,gB;8BAAA,iCnGmHgF,  
mBmGnH1E,wBnGmH0E,EmGnH1D,oBnGmH0D,EmGnH/C,wBAAM,OnGmHyC,CAAkC,WmGnH1H,O;kBAA  
A,2C;uBAAA,yB;cAAA,Q;;cAhBA,OAgBA,a;;;;;;K;IAjBY,sF;MAAA,yD;uBAAA,6H;YAAA,S;iBAAA,Q;;  
iBAAA,uB;O;K;8CABpB,wB;MAUuD,qB;QAAA,QAAa,C;MACHe,wBAAwB,KAAxB,C;MAEA,OAAO,SAAS,g  
DAAT,C;K;+BAsBX,Y;MAMyC,OAAA,oBAaC,W;K;IAEvD,2B;MAAA,+B;MAMBI,uBAA4B,WAAO,uBAAP,  
EAAiC,GAAjC,C;MAC5B,2BAAgC,WAAO,SAAP,EAAoB,GAAPB,C;MAGhC,iCAAsC,WAAO,KAAP,EAAiB,  
GAAjB,C;K;oDAtBtC,mB;MAIwD,oBAAM,oBAAO,OAAP,CAAN,C;K;+CAExD,mB;MAIoD,OAAA,OzGzDyC  
,SyGyDnB,oBzGzDmB,EyGyDJ,MzGzDI,C;K;0DyG2D7F,mB;MAI+D,OAAA,OzG/D8B,SyG+DR,wBzG/DQ,Ey  
G+DW,MzG/DX,C;K;gEyGoE7F,mB;MAAgE,OAAA,OzGpE6B,SyGoEP,8BzGpEO,EyGoEkB,MzGpEIB,C;K;;I  
yG8CjG,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;;IA1PA,4C;MAAA,+C;MACkE,kBAAK,OAAL,EAAc,MAAM,  
MAAN,CAAd,C;MADIE,Y;K;IAGA,sC;MAAA,+C;MAC6C,kBAAK,OAAL,EAAc,UAAa,C;MAD7C,Y;K;IA0R  
O,kG;MAAA,kC;MAAA,8C;MAAA,kC;MAAA,kC;MACH,uBAA+B,a;MAI/B,4F;MA0BA,sBAA0C,I;K;+FA9B1  
C,Y;MAAA,2B;K;+FAEI,Y;MAAQ,qBAAA,kB9S8C,CM8SxC,CN9SwC,CM8S9C,C;K;gGAEZ,Y;MAAA,4B;  
K;iEAqBA,mB;MACI,OAAO,MAAa,UAAU,eAAe,MAAK,CAAL,EAAQ,IAAR,C;K;IAStB,oG;MAAA,kC;MAA  
S,uB;K;mJACG,Y;MAAQ,OAAA,kBAAM,O;K;wGACrC,iB;MAAuC,Q;MAAA,eAAA,kBnJVG,CMiVG,KNjVH  
,CMiVH,mBAAgB,E;K;qGAJnE,Y;MACI,IAAI,2BAAJ,C;QACI,yH;;MAKJ,OAAO,kC;K;4CAGf,Y;MACI,OAA  
Y,SAAZ,wBAAy,EAAS,kBAAT,EAAoB,kBAAM,UAAV,GAAqB,8BAAuB,kBAAM,MAA7B,CAArB,GAA8D,  
kBAAM,aAAN,GAAqB,CAArB,IAA9E,EAAsg,wBAAtG,C;K;gEAEhB,iB;MACI,IAAI,QAAc,iBAAN,kBAAM,  
CAAIB,C;QACI,YAAkB,kBAAy,YAAW,KAAX,C;QAC9B,IAAa,KAAT,sBAAiB,KAAR,C;UACI,YAAkB,kB  
AAy,YAAW,QAAQ,CAAR,IAAX,C;UAC9B,IAAa,KAAT,sBAAiB,KAAR,C;YACI,OAAO,QAAQ,CAAR,I;;;  
MAInB,OAAO,QAAQ,CAAR,I;K;IApDiC,2E;MAAA,kC;MAAA,kB;MAAoC,6B;K;mHACrD,Y;MAAQ,OAAA,  
kBAAM,O;K;IACqC,4E;MAAA,qB;QAAE,yBAAK,EAAL,C;O;K;qEAA5E,Y;MAAiD,OAAqB,OAAb,aAAR,oB  
AAQ,CAAa,EAAL,iEAAJ,CAAiB,W;K;wEACvF,iB;MAA4C,Q;MAAA,eAAA,kBnNtU,CMmTJ,KNnTI,CMmT  
V,YAAoB,oBAApB,O;K;wEAE5C,gB;MAIW,IADwB,IACxB,EAQ6C,MAR7C,EAQA,M;MATwB,OAAZ,kBAA  
Y,O;MAAIB,iB;QACN,MAAM,gCAAYB,gCAA6B,IAA7B,oEAAzB,C;;MADb,aAAa,I;MAKb,IAAI,CAAC,qCA  
AwB,MAAxB,EAAGC,IAAhC,CAAL,C;QACI,MAAM,gCAAYB,gCAA6B,IAA7B,qBAAzB,C;MAEV,YAAY,O  
AAO,IAAP,C;MACL,IAAI,SAAS,SAAb,C;QAAwB,a;;QAAU,wBAAW,4DAAX,C;;MAAzC,a;K;;IA5BhB,uD;M  
ACI,sBAAiB,I;MACjB,YAAY,eAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAMB,OAAO,I;MAC1B,YAAY,aAAA,K  
AAM,MAAN,EAAa,sBAAy,CAAZ,IAAb,C;MAEZ,mE;K;IA8DJ,iD;MAM+B,UAKO,MALP,EAoBD,MApBC,E  
AoBD,MApBC,EAIcD,MAjCC,EAIcD,M;MArC1B,YAAY,C;MACZ,aAAa,sB;MAEb,OAAO,QAAQ,WAAy,O  
AA3B,C;QACI,WAAW,wBAAy,YAAZ,EAAY,oBAAZ,Q;QACX,IAAI,SAAQ,EAZ,C;UACI,IAAI,UAAS,WA  
AY,OAAzB,C;YACI,MAAM,gCAAYB,mCAAzB,C;UAEV,MAAO,gBAAO,wBAAy,cAAZ,EAAY,sBAAZ,UAA  
P,C;eACJ,IAAL,SAAQ,EAZ,C;UACH,IAAI,UAAS,WAAy,OAAzB,C;YACI,MAAM,gCAAYB,kCAAzB,C;UA  
EV,IAAI,uBAAy,KAAZ,MAAsB,GAA1B,C;YACI,eAA2B,cAAZ,WAAy,GAAC,qBAAd,EAAc,KAAd,E;YAE3  
B,IAAI,UAAS,QAAb,C;cACI,MAAM,gCAAYB,8DAAzB,C;YACV,IAAI,aAAy,WAAy,OAAxB,IAAkC,uBAAy  
,QAAZ,MAAyB,GAA/D,C;cACI,MAAM,gCAAYB,yDAAzB,C;YAEV,gBAAgB,WzGvLgE,WyGuL1C,KzGvL0C  
,EyGuLnC,QzGvLmC,C;YyGyLhF,MAAO,gBAAO,0BAAA,KAAM,OAAAN,EAAa,SAAb,qDAaKc,EAAzC,C;Y  
ACP,QAAQ,WAAW,CAAX,I;;YAER,IAAI,EAAuB,kBAAK,EAAL,CAAvB,0CAAY,KAAZ,EAJ,C;cACI,MAA  
M,gCAAYB,mCAAzB,C;YAEV,aAAa,KAAM,O;YACnB,iBAA2B,eAAZ,WAAy,EAAe,KAaf,EAAsB,MAAO,K



MACpE,IAAI,CAAC,UAAAL,C;QACI,O3G4MoE,mB2G5M9C,M3G4M8C,C;;Q2G1MpE,OAAO,yBAAC,mBAAS  
,MAAO,OAAhB,IAAd,EAAc,MAAtC,EAA8C,CAA9C,EAAiD,MAAO,OAAxD,EAAgE,UAAhE,C;K;IAGf,mC;  
MAGI,aACa,S3GmN2D,O2GnNhD,K3GmNgD,C;M2GINxE,OAAO,kBAAkB,MAAO,OAAP,KAAe,C;K;IAG5C,  
4B;MAKoD,gCAAU,C;MAAV,U;QAAuB,kBAAR,yB;QAAQ,c;;UIHioDvD,U;UADhB,IAAI,0CAAsB,qBAA1B,  
C;YAAqC,aAAO,I;YAAP,e;;UACrB,+B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CkHjoD4D,aAA  
T,qBIHioDxC,OkHjoDwC,CAAS,CIHioDhE,C;cAAyB,aAAO,K;cAAP,e;;UAC/C,aAAO,I;;QkHloDgE,iB;;MAA  
vB,W;K;IAEpD,gD;MASiD,0B;QAAA,aAAsB,K;MAOxC,Q;MAN3B,IAAI,iBAAJ,C;QAAkB,OAAO,a;MACzB,I  
AAI,aAAJ,C;QAAMB,OAAO,K;MAC1B,IAAI,CAAC,UAAAL,C;QAAiB,OAAO,kBAAQ,KAAR,C;MAExB,IAAI,  
SAAK,OAAL,KAAe,KAAM,OAazB,C;QAAiC,OAAO,K;MAEb,OAAL,SAAK,O;MAA3B,iBAAC,CAAd,wB;Q  
ACI,eAAe,qBAAK,KAAL,C;QACf,gBAAGB,iBAAM,KAAN,C;QACb,IAAI,CAAU,SAAT,QAAS,EAAO,SAA  
P,EAAkB,UAAIB,CAAd,C;UACI,OAAO,K;;MAIf,OAAO,I;K;IAIX,sF;MACkH,0B;QAAA,aAAsB,K;MACpI,oC  
AAkB,UAAIB,EAA8B,KAA9B,EAAqC,WAArC,EAAkD,MAAlD,EAA0D,UAA1D,C;K;IAGJ,+B;MAYI,OrGm  
MmD,mBAAS,CqGnM5D,G3GiJ4F,oB2GjJzD,C3GiJyD,E2GjJtD,C3GiJsD,CAhE9B,c2GjFrC,G3G8IoD,oB2G9I  
Z,C3G8IY,C2G9I7E,GAAyE,S;K;IAG7E,iC;MASI,OrGuLmD,mBAAS,CqGvL5D,G3GqI4F,oB2GrIzD,C3GqIyD,  
E2GrItD,C3GqIsD,CA3C9B,c2G1FrC,G3GkIoD,oB2GIIZ,C3GkIY,C2GII7E,GAAyE,S;K;IAG7E,8B;MAOiB,IAA  
N,I;MxH/FP,IAAI,EwH8FI,KAAK,CxH9FT,CAAJ,C;QACI,cwH6Fc,oD;QxH5Fd,MAAM,gCAAYB,OAAQ,WAA  
jC,C;;MwH6FH,QAAM,CAAN,C;aACH,C;UAAK,S;UAAAL,K;aACA,C;UAAU,OAAL,SAAK,W;UAAV,K;;UAEl  
,aAAa,E;UACb,IAAI,ErGgKoC,qBAAU,CqGhK9C,CAAJ,C;YACI,QAAQ,SAAK,W;YACb,YAAY,C;YACZ,OA  
AO,IAAP,C;cACI,IAAI,CAAC,QAAU,CAAX,MAAiB,CAArB,C;gBACI,UAAU,C;;cAEd,QAAQ,UAAW,C;cAC  
nB,IAAI,UAAAS,CAAb,C;gBACI,K;;cAEJ,KAAK,C;;;UAGb,OAAO,M;;MAnBf,W;K;IAwBJ,4D;MAOqE,0B;QA  
AA,aAAsB,K;MACvF,O3G2GiG,kB2G3GnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAAgB,K  
AAhB,GAA2B,IAA1D,C3G2GmF,E2G3GIB,6BAAM,iCAAwB,QAAxB,C3G2GyC,C;K;I2GzGrG,4D;MAM+D,0B  
;QAAA,aAAsB,K;MACjF,O3GkGiG,kB2GIGnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,  
UAAJ,GAAgB,KAAhB,GAA2B,IAApE,C3GkGmF,E2GIGA,oBAAR,OAAQ,C3GkGA,C;K;I2GhGrG,iE;MAC0E,  
0B;QAAA,aAAsB,K;MAC5F,O3G8FiG,kB2G9FnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAA  
gB,IAAhB,GAA0B,GAAzD,C3G8FmF,E2G9FpB,6BAAM,iCAAwB,QAAxB,C3G8Fc,C;K;I2G5FrG,iE;MACoE,0  
B;QAAA,aAAsB,K;MACtF,O3G0FiG,kB2G1FnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,  
UAAJ,GAAgB,IAAhB,GAA0B,GAAne,C3G0FmF,E2G1FF,oBAAR,OAAQ,C3G0FE,C;K;I4GtQrG,kD;MAEI,IA  
AI,gBAAJ,C;QAAsB,MAAM,6BAAyB,qCAAK,C,QAAQ,CAAR,IAAIC,CAAzB,C;MAC5B,OAAO,CAAC,IAAD,  
I;K;IAGX,iF;MAQI,IAAI,EAAc,KAAT,oBAAiB,KAAjB,KAA2B,SAAS,QAAxC,C;QACI,OAAO,UAAU,CAAV,  
EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,UAAU,kBAAO,KAAP,C1GyBgC,I;M0GxB1C,IAAI,EAAQ,KAAR,kB  
AAgB,KAAhB,CAAJ,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAO,SAAW,C  
AAC,OAAS,IAAV,KAAqB,EAAhC,IAAwC,MAAQ,I;K;IAG3D,yE;MAQI,IAAI,SAAU,EAAV,MAAkB,CAAIB,I  
AAuB,SAAS,QAAP,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAa,C  
AAP,KAAO,C;MACzB,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAo  
B,gBAApB,C;;MAEX,OAAQ,SAAU,CAAX,GAAKB,KAAIB,GAA4B,I;K;IAGvC,yE;MASI,IAAI,SAAS,QAAb,C  
;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,YAAY,KAAa,CAAP,KAAO,C;MACzB,  
IAAI,SAAU,EAAV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;UAEl,OAAO,UAAU,CAA  
V,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,EAAR,C;QACH,IAAI,SAAU,GAAV,M  
AAkB,GAAtB,C;UAEl,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,GAAV,MA  
AkB,GAAtB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UA  
Aa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QA  
AQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,  
EAAoB,gBAApB,C;;MAGX,OAAQ,SAAU,EAAx,GAAoB,SAAU,CAA9B,GAAqC,KAAR,C,GAA+C,O;K;IAG1  
D,yE;MASI,IAAI,SAAS,QAAb,C;QACI,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGJ,YAAY,KAAa,  
CAAP,KAAO,C;MACzB,IAAI,SAAU,EAAV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GAAV,KAAkB,GAAtB,C;U  
AEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,CAArB,C;QAC  
H,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;UAEl,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,



IAAI, SAAU, EAAV, IAAgB, CAAPB, C; QACH, OAAO, UAAU, CAAV, EAAa, KAAb, EAAoB, gBAAPB, C; aACJ, IA  
AI, SAAU, GAAV, MAAkB, GAAtB, C; QACH, OAAO, UAAU, CAAV, EAAa, KAAb, EAAoB, gBAAPB, C;; MAGX, IA  
AI, SAAQ, CAAR, UAAa, QAAjB, C; QACI, OAAO, UAAU, CAAV, EAAa, KAAb, EAAoB, gBAAPB, C;; MAEX, YAA  
Y, KAAiB, CAAX, QAAQ, CAAR, IAAW, C; MAC7B, IAAI, SAAU, GAAV, MAAkB, GAAtB, C; QACI, OAAO, UAAU,  
CAAV, EAAa, KAAb, EAAoB, gBAAPB, C;; MAGX, IAAI, SAAQ, CAAR, UAAa, QAAjB, C; QACI, OAAO, UAAU, CA  
AV, EAAa, KAAb, EAAoB, gBAAPB, C;; MAEX, YAA Y, KAAiB, CAAX, QAAQ, CAAR, IAAW, C; MAC7B, IAAI, SA  
AU, GAAV, MAAkB, GAAtB, C; QACI, OAAO, UAAU, CAAV, EAAa, KAAb, EAAoB, gBAAPB, C;; MAEX, OAAQ, S  
AAU, EAAX, GAAoB, SAAU, EAA9B, GAAuC, SAAU, CAAjD, GAAwD, KAAxD, GAAkE, O; K;;; IAmB7E, oE; MAk  
B0B, UAGJ, MAHI, EAKJ, MALI, EAMJ, MANI, EASJ, MATI, EAUI, MAVI, EAWJ, MAXI, EAjBA, MAhBA, EAiBA,  
MAjBA, EAkBA, MAiBA, EAoBA, MApBA, EAqBA, OArBA, EAsBA, OAtBA, EAuBA, O; MzH9JtB, IAAI, EyHgII, cA  
Ac, CAAd, IAAMb, YAA Y, MAAO, OAAtC, IAAGD, cAAc, QzHhIIE, CAAJ, C; QACI, cAda, qB; QAeb, MAAM, gCAAy  
B, OAAQ, WAAjC, C;; MyHgIV, YAA Y, cAAU, CAAC, WAAW, UAA X, IAAD, IAA0B, CAA1B, IAAV, C; MACZ, gBA  
AgB, C; MACHb, gBAAgB, U; MAEhB, OAAO, YAA Y, QAA nB, C; QACI, WAAW, mBAAO, gBAAP, EAAO, wBAAP,  
Q1GzH2B, I; Q0G2HIC, WAAO, GAAP, C; UACI, MAAM, kBAAN, EAAM, 0BAAN, YAA0B, OAAAL, IAAK, C; eAC9B  
, WAAO, IAAP, C; UACI, MAAM, kBAAN, EAAM, 0BAAN, YAA4C, OAArB, QAAS, CAAV, GAAgB, GAAM, C; UAC  
5C, MAAM, kBAAN, EAAM, 0BAAN, YAA+C, OAAxB, OAAS, EAAV, GAAMb, GAAM, C; eAEnD, WAAO, KAAP, I  
AAiB, QAAQ, KAAzB, C; UACI, MAAM, kBAAN, EAAM, 0BAAN, YAA6C, OAAtB, QAAS, EAAV, GAAiB, GAAM,  
C; UAC7C, MAAM, kBAAN, EAAM, 0BAAN, YAAuD, OAA/B, QAAS, CAAV, GAAiB, EAA1B, GAA2B, GAAM, C; U  
ACvD, MAAM, kBAAN, EAAM, 0BAAN, YAA+C, OAAxB, OAAS, EAAV, GAAMb, GAAM, C;; UAG/C, gBAAgB, u  
BAAuB, MAAvB, EAA+B, IAA/B, EAAqC, SAArC, EAAGD, QAAhD, EAA0D, gBAA1D, C; UACHb, IAAI, aAAa, CA  
AjB, C; YACI, MAAM, kBAAN, EAAM, 0BAAN, YAAqB, 0BAA0B, CAA1B, C; YACrB, MAAM, kBAAN, EAAM, 0B  
AAN, YAAqB, 0BAA0B, CAA1B, C; YACrB, MAAM, kBAAN, EAAM, 0BAAN, YAAqB, 0BAA0B, CAA1B, C;; YAEr  
B, MAAM, kBAAN, EAAM, 0BAAN, YAAkD, OAA3B, aAAc, EAaf, GAAsB, GAAM, C; YACID, MAAM, mBAAN, E  
AAM, 2BAAN, aAA6D, OAArC, aAAc, EAaf, GAAuB, EAAXB, GAAiC, GAAM, C; YAC7D, MAAM, mBAAN, EAAM  
, 2BAAN, aAA4D, OAApC, aAAc, CAaf, GAAsB, EAavB, GAAgC, GAAM, C; YAC5D, MAAM, mBAAN, EAAM, 2B  
AAN, aAAoD, OAA7B, YAAc, EAaf, GAAwB, GAAM, C; YACpD, 6B;;; MAMhB, OAAW, KAAM, OAAAN, KAAc, SA  
AIB, GAA6B, KAA7B, GAA8C, UAAN, KAAM, EAAO, SAAP, C; K;;; IAQzD, mE; MAiByB, Q; MzH9LrB, IAAI, EyHw  
LI, cAAc, CAAd, IAAMb, YAA Y, KAAM, OAArC, IAA6C, cAAc, QzHxL/D, CAAJ, C; QACI, cAda, qB; QAeb, MAAM, g  
CAAyB, OAAQ, WAAjC, C;; MyHwLV, gBAAgB, U; MACHb, oBAAoB, sB; MAEpB, OAAO, YAA Y, QAA nB, C; QAC  
I, WAAW, KAAMb, CAAb, gBAAa, EAAb, wBAAa, O; QAE1B, YAAQ, CAAR, C; UACI, aAAc, gBAA Y, OAAAL, IAAK,  
CAAZ, C; aACIB, YAAS, CAAT, KAAc, EAAd, C; UACI, WAAW, eAAe, KAAf, EAAsB, IAAtB, EAA4B, SAA5B, EAA  
uC, QAAvC, EAAiD, gBAAjD, C; UACX, IAAI, QAAQ, CAAZ, C; YACI, aAAc, gBAAO, gBAAP, C; YACd, yBAAa, CA  
AC, IAAD, IAAb, K;;; YAEA, aAAc, gBAA Y, OAAAL, IAAK, CAAZ, C; YACd, wBAAa, CAAb, I;;; eAGR, YAAS, CAAT, K  
AAc, EAAd, C; UACI, aAAW, eAAe, KAAf, EAAsB, IAAtB, EAA4B, SAA5B, EAAuC, QAAvC, EAAiD, gBAAjD, C; U  
ACX, IAAI, UAAQ, CAAZ, C; YACI, aAAc, gBAAO, gBAAP, C; YACd, yBAAa, CAAC, MAAD, IAAb, K;;; YAEA, aAAc  
, gBAA Y, OAAAL, MAAK, CAAZ, C; YACd, wBAAa, CAAb, I;;; eAGR, YAAS, CAAT, KAAc, EAAd, C; UACI, aAAW, eA  
Ae, KAAf, EAAsB, IAAtB, EAA4B, SAA5B, EAAuC, QAAvC, EAAiD, gBAAjD, C; UACX, IAAI, UAAQ, CAAZ, C; YA  
CI, aAAc, gBAAO, gBAAP, C; YACd, yBAAa, CAAC, MAAD, IAAb, K;;; YAEA, WAA Y, MAAD, GAAQ, KAAR, IAAq  
B, EAARb, GAA2B, K; YACtC, UAAW, SAAS, IAAV, GAAoB, K; YAC9B, aAAc, gBAA Y, OAAAL, IAAK, CAAZ, C; YA  
Cd, aAAc, gBAAW, OAAJ, GAAI, CAAX, C; YACd, wBAAa, CAAb, I;;; UAIJ, UAAU, CAAV, EAAa, SAAb, EAAwB, gB  
AAxB, C; UACA, aAAc, gBAAO, gBAAP, C;;; MAK1B, OAAO, aAAc, W; K; ICtQzB, uC; MAU2D, OAAwB, CAAXB, 2B  
AAwB, mBAAS, SAAT, C; K; IAEnF, oC; MAKI, OAAQ, OAAW, mBAAL, SAAK, CAAX, C; K; IAGZ, 6C; MAMI, IAAI,  
cAAS, SAAb, C; QACI, iBAAsB, SAAY, Y; QACIC, IAAI, kBAAJ, C; UACS, SAAL, eAA+B, iBAAc, SAAd, E;;; UAE/B, U  
AAW, WAAI, SAAJ, C;;; K; IAUnB, 6C; MAC4B, UAAjB, M; MAAP, OAAO, WAAiB, OAAZ, SAAY, YAAjB, 4CAA+D  
, W; K; IAI9E, iC; MACI, gBAAqB, sB; MACrB, iBAAsB, E; MACtB, kBAA+B, E; MAC/B, uBAAiC, C; K; uDAEjC, qB; M  
ACc, qBAAV, SAAU, EAAC, EAAd, EAAB, EAALB, C; MACV, OAAO, aAAO, W; K; gDAGIB, qB; MAA6D, gBAAR, c;  
MAAQ, c;; Qzlu4Y7C, Q; QAAbB, wBAAgB, SAAhB, gB; UAAgB, cAAA, SAAhB, M; UAAsB, IAAC, OyIv4Y+B, czlu4  
Y7C, C; YAAwB, aAAO, I; YAAP, e;;; QAC9C, aAAO, K;;; MyIx4Y8C, iB; K; sDAErD, wC; MACI, KAAK, qBAAL, SAA

K,EAAC,MAAd,EAASB,SAAtB,CAAL,C;QAAyC,M;MAEzC,YAAY,SAAK,M;MACjB,OAAO,aAAP,C;QACI,K  
AAM,qBAAN,KAAM,EAAC,MAAd,EAASB,aAAtB,CAAN,C;UAA8C,M;QAC9C,QAAQ,KAAM,M;;K;sDAItB,  
wC;MASgB,IAAiB,IAAjB,EA2BE,M;MANCd,aAAO,gBAAO,MAAP,CAAe,gBAAO,SAAP,C;MACtB,gBAAgB,  
SAAK,W;MACrB,IAAI,eAAQ,SAAR,CAAJ,C;QACI,aAAO,gBAAO,kCAAP,CAA2C,gBAAO,SAAP,CAAKB,gB  
AAO,KAAP,C;QACpE,OAAO,K;;MAEH,cAAY,MAAK,SAAL,C;MAEpB,YAAY,CAAiB,OAAZ,SAAY,MAAjB  
,2D;MACZ,IAAI,aAAJ,C;QvHyBG,SuHxBwB,WAAN,KAAM,EAQ,SAAR,C;QAAvB,iBAoD,KAAK,CAAT,  
GAAY,CAAZ,GAAMB,KAAe,gBAaf,I;QACnE,IAAI,eAAc,CAAIb,C;UAAqB,aAAO,gBAAO,SAAP,CAAKB,g  
BAAO,IAAP,C;QAC9C,IAAI,evG8MoC,YAAU,CuG9MID,C;UACI,kBAAW,K;UACX,uBAAGB,U;;UAEhB,QA  
AQ,wBAAiB,KAAjB,EAAwB,UAAxB,C;;QAEZ,IAAI,MvGgNuC,UAAS,CuGhNpD,C;UAEuB,U;UAAA,IAAI,e  
AAc,CAAIb,C;YAAA,SAAQb,C;;YxG0+BpC,U;YADhB,YAAY,C;YACI,oBwG1+B+C,SxG0+B/C,C;YAAhB,O  
AAGB,gBAAhB,C;cAAGB,sC;cAAM,IwG1+BgE,UxG0+BiD,oBwG1+BkD,MAAK,ExG0+BrE,C;gBAAwB,qB;;  
YwG1+Bf,SA4B,IxG2+BpD,KwG3+BoD,I;;UAA/C,yB;U1GyrCC,kB;UADb,YAAY,C;UACC,S0GxrCK,aAN,  
KAAM,C1GwrCL,W;UAAb,OAAa,gBAAb,C;YAAa,wB;Y0GvrCG,I1GurCU,oBAAMb,cAAnB,EAAmB,sBAAn  
B,U0GvrCN,gBAAJ,C;cAA2B,aAAO,uB;YACIC,aAAO,gB1GsrCgC,I0GtrChC,CAAA,gBAAO,IAAP,C;;;UAGxB  
,aAAO,gBAAO,KAAP,CAAc,gBAAO,IAAP,C;;;QAGzB,aAAO,gBAAO,SAAP,CAAKB,gBAAO,IAAP,C;;MAG7  
B,iBAAiB,mC;MACjB,IpIyHoD,CoIzHhD,UpIyHiD,UoIzHrD,C;QACI,uBAAuB,SAAS,M;QACtB,8B;QAAV,OA  
AU,gBAAV,C;UAAU,qB;UACJ,qBAAF,CAAe,EAAC,gBAAd,EAAgC,cAAhC,C;;;MAGV,OAAO,I;K;yDAGX,6  
B;MAIwB,Q;MAHpB,mBAAwB,C;MACxB,gBAAqB,C;MACrB,mBAAwB,C;MACJ,OtHyIjB,MAAO,KsHzIgb,  
eAAS,OAAT,GAAKB,oBAAIb,ItHyIhB,EsHzIiD,KAAM,OAAN,GAAe,UAAf,ItHyIjD,C;MsHzIV,eAAY,CAAZ,  
oB;QACI,QAAQ,iBAAY,iBAAN,KAAM,CAAN,GAAKB,GAAIb,IAAN,C;QACR,IAAI,MAAK,2BAAKB,iBAAT  
,eAAS,CAAT,GAAqB,GAArB,IAAT,CAAT,C;UAA6C,K;QAC7C,IAAI,MAAK,EAAT,C;UACI,8BAAGB,CAAh  
B,I;UACA,eAAe,S;UACf,YAAY,G;;;MAGpB,IAAI,gBAAGB,CAApB,C;QAAuB,OAAO,K;MAC9B,OAAO,eAA  
e,CAAf,IAAoB,iBAAY,iBAAN,KAAM,CAAN,IAAmB,YAAnB,GAAkC,CAAIc,KAAN,MAA+C,EAA1E,C;QA  
CI,8BAAGB,CAAhB,I;MAGJ,OAAa,YAAN,KAAM,EAAS,YAAT,CAAN,IAA+B,cAAW,eAAe,CAAf,IAAX,uC  
AA/B,C,K;;yHC/H+C,Y;MAAQ,W;K;IAEtE,gD;MACkB,UAMP,M;MANO,IAAI,aAAY,CAAhB,C;QACV,Y;;Q  
AEA,UxBuZ8C,MAAW,KwBvZ/C,IxBuZ+C,EwBvZtC,QxBuZsC,C;QwBtZzD,OAAA,IAAO,OxB2UmC,MAAW  
,KwB3UpC,KxB2UoC,CwB3UxC,GAAa,GAAnB,CAAP,GAAiC,GAAjC,GxBwV2C,WwBxVC,KxBwVD,C;;Mw  
B5V/C,kB;MAMO,IxByUuC,MAAW,KwBzU1C,OxByU0C,CwBzU9C,GAAe,MAAnB,C;QAEEmC,SA9B,OAA  
Y,SAAQ,QAAR,C;;QAGpB,exBoU0C,MAAW,KwBpU1C,OxBoUkC,C;QwBnUrD,qBAA8B,QAAy,axBgRC,MA  
AW,MAvCV,YwBzOqB,QxByOrB,CAuCU,CwBhRA,GAAwB,QAAP,C;QAC1C,SAAI,UAAU,CAAd,GAAiB,  
MAAG,cAAPB,GAAyC,c;;MAP7C,a;K;IAWJ,6C;MACI,OAAa,KAAY,gBAAe,OAAf,EAAwB,MAAK,4BAA2B,  
QAA3B,CAAL,EAAXB,C;K;ICtBQ,4C;MAFrC,e;MAEsC,0B;MAFtC,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;M  
AGI,uEAGY,GAHZ,C;MAIA,yEAGa,MAHb,C;MAIA,yEAGa,SAHb,C;MAIA,+DAGQ,KAHR,C;MAIA,+DAGQ  
,MAHR,C;MAIA,2DAGM,MAHN,C;MAIA,yDAGK,OAHL,C;K;;IAxBA,gD;MAAA,yB;MAAA,wC;K;;IAIA,iD;  
MAAA,yB;MAAA,yC;K;;IAIA,iD;MAAA,yB;MAAA,yC;K;;IAIA,4C;MAAA,yB;MAAA,oC;K;;IAIA,4C;MAAA  
,yB;MAAA,oC;K;;IAIA,0C;MAAA,yB;MAAA,kC;K;;IAIA,yC;MAAA,yB;MAAA,iC;K;;IA3BJ,+B;MAAA,4Q;K  
;;IAAA,oC;MAAA,a;AAA,a;UAAA,6C;aAAA,c;UAAA,8C;aAAA,c;UAAA,8C;aAAA,S;UAAA,yC;aAAA,S;UA  
AA,yC;aAAA,O;UAAA,uC;aAAA,M;UAAA,sC;;UAAA,6D;;K;;IAiCA,4D;MAGW,Q;MADP,0BAA2C,iBAAjB,  
UAAW,cAAM,EAAU,UAAW,cAARb,C;MAEvC,0BAASB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,U  
AAW,cAAvC,C;WAC3B,0BAASB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;QACn  
B,Y;MAHZ,W;K;IAOJ,oE;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAARb,C;MAEvC  
,0BAASB,CAAtB,C;QAA2B,sBAA8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAHZ,  
W;K;IAOJ,8D;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAARb,C;MAEvC,0BAASB,C  
AAtB,C;QACI,YAAkD,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,C;QACID,aAAa,eAAQ,KAAR,C;QAET,  
sBAAS,KAAT,GAAKB,KAAIb,E;UAA2B,a;aAC3B,uBAAQ,CAAR,C;;;aAIR,0BAASB,CAAtB,C;QAA2B,iBAA  
8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAXZ,W;K;;;ICvCJ,+B;MAAA,mC;  
MAWiB,wB;MANT,aAAyB,OAAO,OAAQ,KAAI,WAAY,IAAG,OAAO,SAAS,IAAyD,CAAC,CAAC,OAAO,S

AAS,K;MADrG,sBAGQ,MAHR,GAIQ,iBAAa,OAAb,CAJR,GAMQ,qBACK,OADF,OAAO,IAAK,KAAI,WAAAn  
B,GAAiC,IAAjC,GAAwC,UAAxC,4GAIO,+B;K;4CAGf,Y;MAA+C,OAAA,mBAAa,U;K;wDAC5D,oB;MAAqE,  
OAAA,mBAAa,qBAAY,QAAZ,C;K;8DACIF,wB;MAA8F,OAAA,mBAAa,2BAAkB,GAAIB,EAAuB,OAAvB,C;  
K;0DAE3G,8B;MACI,OAAA,mBAAa,uBAAC,QAAd,EAAwB,QAAxB,C;K;;;IATrB,2C;MAAA,0C;QAAA,yB;;  
MAAA,mC;K;IA+B2B,+B;MAAC,wB;K;qCAExB,Y;MAAwC,8CAAc,cAAQ,SAAtB,C;K;iDACxC,oB;MAEmB,  
IAAS,I;MIHmDrB,Q0HnDH,cAAQ,QAAO,eAAS,OAAT,QAAS,gBAAT,uBAAP,C;MACI,c5IICT,EAAI,CAAJ,C  
;M4IkCb,Y5IiEIB,EAAI,CAAJ,C;M4InEH,OAeUc,aAAR,OAAQ,qCAAR,aAAiD,aAAN,KAAM,yCAAjD,C;K;  
uDAEnC,wB;MAEmB,IAAI,IAAJ,EACQ,M;mBADR,eAAI,OAAJ,GAAI,gBAAJ,uB;MAAV,S5IvCF,OAAI,CAAJ  
,C;M4IsCM,S5I6DN,OAAI,CAAJ,C;mB4I5DY,eAAQ,SAAR,OAAQ,gBAAR,yB;MAAV,S5IvCF,OAAI,CAAJ,C;  
M4IuCM,S5I4DN,OAAI,CAAJ,C;M4I3DH,OAAO,CAAK,OAAM,EAAN,IAAY,OAAM,EAAtB,GAA0B,gCAAS  
,KAAAnC,GAAuD,aAAT,KAAK,EAAI,qCAAXD,cAAsG,aAAT,KAAK,EAAI,yCAAtG,C;K;mDAGX,8B;MAEK,I  
AAS,I;MIHuCP,Q0HvCF,eAAS,OAAT,QAAS,gBAAT,uB;MAA0C,c5I7CxC,EAAI,CAAJ,C;M4I6CiD,Y5IsDjD,  
EAAI,CAAJ,C;MUKIBW,uB;MAAP,eAAuB,6B;MkItoBtB,oBAAoB,YAA Y,U1BuPO,Y0BvPqB,6D1BuPrB,C0Bv  
PnB,C;MAH5B,8CAIQ,CAAgB,aAAhB,EAAiD,SAAd,aAAc,CAAlB,GAA8B,QAAQ,QAAtC,GAAoD,GAAf,C  
AJR,C;K;sCASJ,Y;MAAkC,qC;K;;IAKF,4C;MAAC,8B;K;6CAEjC,Y;MAA6B,OAAA,gBAAY,M;K;8CAEzC,Y;  
MAAwC,8CAAc,aAAAd,C;K;0DACxC,oB;MAAwE,IAAS,I;MAAnB,OIIgCZ,akIhCa,iBAAS,QAAS,OAAT,QAAS  
,gBAAT,oCAAT,CIIgCb,4B;K;gEkI9BID,wB;MACc,IAAI,IAAJ,EACQ,M;MADIB,UAAU,QAAI,OAAJ,GAAI,g  
BAAJ,oC;MACV,UAAU,QAAQ,SAAR,OAAQ,gBAAR,sC;MACV,OAAW,QAAO,GAAIB,GAAuB,gCAAS,KA  
AhC,GII2B8C,akI3BH,MAAM,GII2BH,4B;K;4DkIxBiD,8B;MAC8B,IAAS,I;MAAnC,8CAAc,YAA Y,SAAS,OA  
AT,QAAS,gBAAT,wCAA6B,QAAS,0DAAID,CAAd,C;K;+CAEJ,Y;MAAkC,2C;K;;;IAGtC,6B;MAAA,iC;K;yCA  
GI,Y;MAA6B,OAAe,U;K;0CAE5C,Y;MAAwC,8CAAc,aAAAd,C;K;sDACxC,oB;MAAwE,IAAS,I;MAAnB,OIIYZ  
,akIZa,iBAAS,QAAS,OAAT,QAAS,gBAAT,oCAAT,CIIyB,4B;K;4DkIVID,wB;MACc,IAAI,IAAJ,EACQ,M;MA  
DIB,UAAU,QAAI,OAAJ,GAAI,gBAAJ,oC;MACV,UAAU,QAAQ,SAAR,OAAQ,gBAAR,sC;MACV,OAAW,QA  
AO,GAAIB,GAAuB,gCAAS,KAhC,GII08C,akIPH,MAAM,GII0H,4B;K;wDkIJD,8B;MAC8B,IAAS,I;MAAnC  
,8CAAc,YAA Y,SAAS,OAAT,QAAS,gBAAT,wCAA6B,QAAS,0DAAID,CAAd,C;K;2CAEJ,Y;MAAkC,+B;K;;;I  
AjBtC,yC;MAAA,wC;QAAA,uB;;MAAA,iC;K;IAoBA,4B;MAA8D,IAAO,QAApB,KAAoB,CAAP,C;QAAGB,M  
AAM,gCAAyB,uCAAzB,C;MAAnC,Y;K;ICIHjD,gD;MAQ+B,kBAApB,wBAAc,IAAd,C;MAA0B,I3HgEjC,a;M2  
HhEA,O3HiEO,W;K;I2H9DX,gD;MAQqD,kBAA1B,gBAAhB,sCAAgB,EAAC,IAAd,EAAoB,IAApB,C;MAAiC,s  
B3HoEID,W2HpEkD,C;MAAxD,O3HqEO,W;K;I4HzFX,yC;MAEKD,8B;MAAA,OCGN,aDHwB,yBAAa,QAAb,  
mCCGxB,C7G+xBgC,sB;K;I4GhyB5E,2C;M9IugIW,kBAA Y,gB;MAoGH,Q;MAAhB,wB8IpmIqB,U9IomIrB,gB;  
QAAGB,c8IpmIK,U9IomIrB,M;QAAsB,IAAI,C8IpmIkB,sB9IomIP,O8IpmIO,C9IomItB,C;UAAyB,WAA Y,WAAI  
,OAAJ,C;;M8IpmI3D,qB9IqmIO,W;M8IpmIP,IzIkNwD,CyIINpD,czIkNqD,UyIINzD,C;Q5GgKuC,U;Q4G/JnC,qB  
5G+JyD,OAAtB,+B4G/Jd,mB5G+Jc,uBAAsB,CAAO,W;QoGkO7C,kBAAhB,sB;QQ/XC,0C;QACA,IAAI,E5G8Q  
oC,0BAAU,C4G9Q9C,CAAJ,C;UACI,2BAAO,GAAP,C;;QAEW,sCAAa,GAAb,C;QALnB,sB5H4DG,WoHoUqC,  
W;QzXxC,OAAO,I;;MAGX,OAAO,K;K;IAGX,8C;MAOmB,c;;Q9I45YC,Q;QAAhB,wB8I55YI,U9I45YJ,gB;U  
AAgB,c8I55YZ,U9I45YJ,M;UAAsB,I8I55YD,sB9I45Ye,O8I55Yf,C9I45YC,C;YAAwB,aAAO,I;YAAP,e;;QAC9  
C,aAAO,K;;;M8I75YP,e;QACI,kBAA6B,MAAX,UAAW,C;Q5GyIM,U;Q4GxIb,a5GwImC,OAAtB,+B4GxIvB,m  
B5GwIuB,uBAAsB,CAAO,W;Q4GxIX,kBC/BjB,aD+BD,MC/BC,C7Gg1C6C,uBAAzB,CAAYB,C;QbjmB9E,kBA  
AS,gB;QA2FA,U;QAAA,+B;QAAhB,OAAgB,gBAAhB,C;UAAgB,6B;UAAM,IyH3yB4C,4BzH2yB9B,SyH3yB8  
B,CzH2yB5C,C;YAAwB,WAA Y,WAAI,SAAJ,C;;QyH3yBtD,sBAAMf,eZ4yBhF,WyH5yBgF,EAAa,GAAb,C;Q  
ACnF,OAAO,I;;MAGX,OAAO,K;K;IEnCP,iC;MAAQ,8BAAY,IAAK,UAAjB,IAA8B,uBAAY,IAAK,mB;K;IAO  
vD,oC;MAAQ,8BAAY,IAAK,a;K;ICZ7B,4B;MAGI,OAAO,yBAAP,C;QACI,sBAAY,mCAAZ,C;;K;IAIR,uC;MA  
OI,sBAAY,sCAAgB,gBAAE,IAAf,CAA5B,C;MACA,OAAO,S;K;ICbP,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO  
,K;K;IAKX,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;iHCoBf,sJ;MAEyC,qB;QAAA,QAaKB,I;MAAM,qB;  
QAAA,QAaKB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,yB;QAAA,YAAsB,I;MAAM,kC;QAAA,qBAA+B,I;MA  
AM,qC;QAAA,wBAaK,C;MAAO,+C;QAAA,kCAA4C,K;MAAO,4C;QAAA,+BAAyC,K;MACtT,QAAQ,E;MA  
CR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAi  
B,S;MACjB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iCAAF,IAAuC,+B;M

ACvC,EAAE,8BAAF,IAAoC,4B;MACpC,OAAO,C;K;+GAw0BX,wD;MAEwC,6B;QAAA,gBAAyB,E;MAAI,uB  
;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/I,QAAQ,E;MACR,EAAE,  
eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;M  
AChB,OAAO,C;K;6EA6CX,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7D,mC;MAEoE,UAAy,KAAZ,IAAqB,K;K;6  
EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEqE,UAAy,KAAZ,IAAqB,K;K;6EAuB1F,4B;MAEq  
E,iBAAY,KAAZ,C;K;6EAErE,mC;MAE4E,UAAy,KAAZ,IAAqB,K;K;6EAuBjG,4B;MAE+D,iBAAY,KAAZ,C;  
K;6EAE/D,mC;MAEsE,UAAy,KAAZ,IAAqB,K;K;6EAuB3F,4B;MAEgE,iBAAY,KAAZ,C;K;6EAEhE,mC;MAE  
uE,UAAy,KAAZ,IAAqB,K;K;6EAuB5F,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7D,mC;MAEoE,UAAy,KAAZ,I  
AAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEqE,UAAy,KAAZ,IAAqB,K;K;6EAuB1  
F,4B;MAEiE,iBAAY,KAAZ,C;K;6EAEjE,mC;MAEwE,UAAy,KAAZ,IAAqB,K;K;8EAuB7F,4B;MAEkE,iBAA  
Y,KAAZ,C;K;6EAEIE,mC;MAEyE,UAAy,KAAZ,IAAqB,K;K;6GC3oC9F,wD;MAEqC,6B;QAAA,gBAA+B,I;M  
AAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpJ,QAAQ,E;MAC  
R,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IA  
AgB,Q;MACHB,OAAO,C;K;mIAiCX,+B;MAEgD,mC;QAAA,sBAAGC,K;MAC5E,QAAQ,E;MACR,EAAE,qBA  
AF,IAA2B,mB;MAC3B,OAAO,C;K;4EC9CX,4B;MAEgE,iBAAY,KAAZ,C;K;4EAgChE,4B;MAEyE,iBAAY,KA  
AZ,C;K;4EAIzE,4B;MAEmE,iBAAY,KAAZ,C;K;4EAYnE,4B;MAE0E,iBAAY,KAAZ,C;K;oIC7a1E,4H;MAE  
8C,qB;QAAA,QAAiB,E;MAAI,6B;QAAA,gBAAgC,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,iC;QAAA,oBAA2  
D,E;MAAW,qC;QAAA,wBAmjvJ,U;;ManJqO,+B;QAAA,kBAmjro,U;;ManJ6S,4B;QAAA,eAA+B,S;MAC3a,Q  
AAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,eAAF,IAAqB,a;MACrB,EAAE,mBAAF,IAAyB,iB;MACzB,  
EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,c  
AAF,IAAoB,Y;MACpB,OAAO,C;K;wIAYX,mC;MAEgD,2B;QAAA,cAAuB,E;MAAI,0B;QAAA,aAAsB,E;MAC  
7F,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,YAAF,IAAkB,U;MACIB,OAAO,C;K;8HAKEX,+D;  
MAEqG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;M  
ACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,I  
AAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;4HAwBX,iE;MAE0C,4B;QAAA,eAAwB,E;MAA  
I,wB;QAAA,WAAyB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,  
K;MAC/K,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,SAAF,IAA  
e,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;sGAUqE,qB;MAAQ,O  
AAW,U;K;sGAEnB,qB;MAAQ,OAAW,U;K;4GAehB,qB;MAAQ,OAAc,a;K;wGAS1B,qB;MAAQ,OAAy,W;K;0  
HAEX,qB;MAAQ,OAAqB,oB;K;kGASnD,qB;MAAQ,OAAQ,Q;K;oGAehB,qB;MAAQ,OAAU,S;K;sGAEjB,qB;  
MAAQ,OAAW,U;K;wHAEV,qB;MAAQ,OAAoB,mB;K;wHAE5B,qB;MAAQ,OAAoB,mB;K;kHAE/B,qB;MAA  
Q,OAAiB,gB;K;kHAEzB,qB;MAAQ,OAAiB,gB;K;oHASd,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAk  
B,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;wIAehB,qB;MAAQ,OAA4B,2B;K;4FC1MnI,uD;MAE8B,oB;QAAA,  
OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;Q  
AAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,S  
AAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGAuBX,sE;  
MAEiC,6B;QAAA,gBAA8B,I;MAAM,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAA  
oB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvL,QAAQ,E;MACR,EAAE,eAAF,IAAq  
B,a;MACrB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YA  
AF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGA8DX,8U;MAEiC,uB;QAAA,UAAgB,C;  
MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SA  
AiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SA AkB,I;MAAM,uB;QA  
AA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SA AmB,K;MAAO,uB;QAAA,UAAoB,K;MAA  
O,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA  
2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;Q  
AAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;M  
AAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq  
B,K;MAC3wB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O

;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EA AE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB ,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBA AF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;M ACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K; wGAgDX,kQ;MAEoC,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAA O,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAA uB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;Q AAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K; MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAu B,K;MAAO,wB;QAAA,WAAqB,K;MAC7IB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAg B,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EA AE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,I AaqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MA CrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,Q AAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAAgB,Q;MA ChB,OAAO,C;K;kGAsCX,iX;MAEiC,sB;QAAA,SAakB,G;MAAK,sB;QAAA,SAakB,G;MAAK,sB;QAAA,SAA kB,G;MAAK,yB;QAAA,YAAkB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAA A,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B ;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAakB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;M AAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mB AA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B; QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K; MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAA oB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACr2B,QAAQ,E;MACR,EAAE,QAAF,IAA c,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,WAAF,IAAiB,S;MACjB,EAAE, SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EA AE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;M ACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IA Ae,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;M ACiB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,o BAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0 B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE, YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;kGA2BX,0E;MAEiC,oB;QAAA,OAAgB, E;MAAI,2B;QAAA,cAAwB,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAA oB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtM,QAAQ,E;MACR,EAAE,MAAF,IAA Y,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE, SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACiB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wGAmDX,4 S;MAEoC,mB;QAAA,MAAe,E;MAAI,oB;QAAA,OAAgB,E;MAAI,wB;QAAA,WAAiB,C;MAAG,sB;QAAA,SA AmB,K;MAAO,2B;QAAA,cAAwB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;Q AAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K; MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,k BAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO, kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;M AAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjtB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MA CX,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,I

AAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;8GAuBX,6D;MAEuC,oB;QAAA,OAGB,E;MAAI,oB;QAAA,OAGB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7K,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wECnBX,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,2B;MAEgG,iBAAY,IAAZ,C;K;wEAWBhG,oC;MAE+F,UAA,Y,KAAZ,IAAqB,M;K;wEAmFpH,2B;MAEqE,iBAAY,IAAZ,C;K;wEAERe,kC;MAE2E,UAA,Y,IAAZ,IAAoB,K;K;wEAssC/F,4B;MAEyE,iBAAY,KAAZ,C;K;wEA0BzE,4B;MAEyE,iBAAY,KAAZ,C;K;wEAsBzE,4B;MAEuE,iBAAY,KAAZ,C;K;wEAyBvE,4B;MAE6E,iBAAY,KAAZ,C;K;2FA4C7E,gD;MAEiC,qB;QAAA,QAAiD,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACIK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uEA+UX,4B;MAEuE,iBAAY,KAAZ,C;K;wEAEvE,2B;MAE6F,iBAAY,IAAZ,C;K;wEAqN7F,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,oC;MAE2F,UAA,Y,KAAZ,IAAqB,M;K;+FAuehH,wD;MAEmC,6B;QAAA,gBAA8B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uGAuIX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+HAYCX,iB;MAEmD,qB;QAAA,QAakB,I;MACjE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+FA0MX,sE;MAEmC,oB;QAAA,OAGB,I;MAAM,wB;QAAA,WA0+G4B,S;MA1+GwB,kB;QAAA,KAAc,E;MAAI,wB;QAAA,WAAoB,I;MAAM,sB;QAAA,SAakB,S;MAAW,uB;QAAA,UAAoB,I;MAAM,qB;QAAA,QAAiB,I;MAAM,oB;QAAA,OAGB,I;MACnP,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,IAAF,IAAU,E;MACV,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qIAGDX,iB;MAEsD,qB;QAAA,QAakB,I;MACpE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+GAKBX,qB;MAE2C,yB;QAAA,YAAmB,S;MAC1D,QAAQ,E;MACR,EAAE,SAAF,IAAe,S;MACf,OAAO,C;K;wEAKCX,4B;MAEqF,iBAAY,KAAZ,C;K;yFAGCrF,4V;MAEGC,4B;QAAA,eAA8B,I;MAAM,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkBB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAakB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAGB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC9yB,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wEAWEX,2B;MAE+D,iBAAY,IAAZ,C;K;iGA2D/D,gD;MAEOC,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACII,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;

qGA2BX,yD;MAEsC,sB;QAAA,SAAkB,E;MAAI,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;  
QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC5J,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,  
QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;M  
AChB,OAAO,C;K;6GAuBX,oD;MAE0C,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,  
aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,  
IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;2FAoFX,kF;MAEi  
C,uB;QAAA,UAAmB,E;MAAI,wB;QAAA,WAAoB,E;MAAI,sB;QAAA,SA Ae,C;MAAG,qB;QAAA,QAAC,C;M  
AAG,qB;QAAA,QAAC,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq  
B,K;MACjN,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc  
,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAA  
F,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;iHAYBX,0D;MAEqE,sB;QAAA,SA Ae,S;MAA  
W,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACzK,QAAQ,E;MACR,  
EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;  
MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wEAmXX,4B;MAEkE,iBAAY,KAAZ,C;K;wEAEIE,2B;M  
AEoE,iBAAY,IAAZ,C;K;wEAUpE,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;wE  
AaxE,4B;MAE+D,iBAAY,KAAZ,C;K;wEAE/D,2B;MAEiE,iBAAY,IAAZ,C;K;mGA0CjE,8G;MAEqC,gC;QAAA  
,mBAoF8C,M;;MApoFe,gC;QAAA,mBAmpFT,S;;MAnpFyE,oC;QAAA,uBA8pFjE,S;;MA9pF6I,2B;QAAA,cAA  
oB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,6B;QAAA,gBAyqFIO,K;;MAxqFvE,QAAQ,E;MACR,EAAE,kBAAF,  
IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,sBAAF,IAA4B,oB;MAC5B,EAAE,aAAF,IAAm  
B,W;MACnB,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,eAAF,IAAqB,a;MACrB,OAAO,C;K;+FAwCX,mF;MAEm  
C,oB;QAAA,OAAa,I;MAAM,sB;QAAA,SAAkB,E;MAAI,2B;QAAA,cAAuB,E;MAAI,sB;QAAA,SAAYC,I;MAA  
M,qB;QAAA,QAA6B,E;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB  
,K;MACxQ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,  
W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YA  
AF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6FA4BX,2B;MAEkC,+B;QAAA,kBAA4B,K;  
MAC1D,QAAQ,E;MACR,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;2FA2DX,iE;MAEiC,wB;QAAA,WAAqB  
,K;MAAO,oB;QAAA,OAAe,C;MAAG,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aA  
AuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,MAAF,  
IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAA  
E,UAAF,IAAgB,Q;MACHB,OAAO,C;K;yFA8FX,6B;MAEgC,oB;QAAA,OA+7E6C,S;;MA/7EL,2B;QAAA,cC12  
He,M;;MDm2HnF,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,OAAO,C;K;  
wEAoDX,0B;MAE+D,iBAAY,GA AZ,C;K;wEAE/D,iC;MAEqE,UAA Y,GA AZ,IAAmB,K;K;+FAoDxF,oF;MAE  
mC,mB;QAAA,MAAe,I;MAAM,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAoB,I;MAAM,mB;QAAA,MAAe  
,E;MAAI,2B;QAAA,cAAwB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,W  
AAqB,K;MACvO,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,UA  
AF,IAAgB,Q;MACHB,EAAE,KAAF,IAAW,G;MACX,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;M  
ACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;iFAwNX,yC;MAE4B,uB;QA  
AA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtG,QAAQ,E;MACR,EAAE,SA  
AF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6FAwBX,iD;M  
AEkC,sB;QAAA,SA Ae,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq  
B,K;MACjI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U  
;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;uGASX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAA  
Q,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAYX,kC;MAE0C,uB;QAAA,UAAoB,K;MAAO,oB;QA  
AA,OAAiB,K;MAAO,uB;QAAA,UAAoB,K;MAC7G,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,MA  
AF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wEAkEX,4B;MAE6D,iBAAY,KAAZ,C;K;wEAU7D  
,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;uGAsCxE,oH;MAEuC,yB;QAAA,YAA  
sB,K;MAAO,0B;QAAA,aAAuB,S;MAAW,6B;QAAA,gBAA0B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,iC;QAA  
A,oBAA8B,S;MAAW,qC;QAAA,wBAaK,C,S;MAAW,+B;QAAA,kBAaK,C,S;MAC1R,QAAQ,E;MACR,EAAE,

WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;  
MACf,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,O  
AAO,C;K;mGAgFX,oB;MAEqC,wB;QAAA,WAAqB,K;MACtD,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAC  
hB,OAAO,C;K;wEA+MX,2B;MAEiE,iBAAY,IAAZ,C;K;2GAKcJE,c;MAEyC,kB;QAAA,KAAgB,S;MACrD,QA  
AQ,E;MACR,EAAE,IAAF,IAAU,E;MACV,OAAO,C;K;2FAuMX,gB;MAGI,QAAQ,E;MACR,EAAE,MAAF,IA  
AY,I;MACZ,OAAO,C;K;wEAgBX,4B;MAEiE,iBAAY,KAAZ,C;K;wEAEjE,oC;MAE4E,iBAAY,aAAZ,C;K;wE  
AuT5E,4B;MAEmE,iBAAY,KAAZ,C;K;uFA2CnE,sB;MAE+B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MA  
AK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAC9F,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE  
,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,OAAO,C;K;qFA0CX,+B;M  
AE8B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;  
MACtG,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,OAAF,IAAa,K;M  
ACb,EAAE,QAAF,IAAc,M;MACd,OAAO,C;K;wEAOX,4B;MAEmE,iBAAY,KAAZ,C;K;yFAiHnE,oB;MAEgC,  
wB;QAAA,WaY2B+C,M;MAx2B3E,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6FAeX,+B;  
MAEkC,oB;QAAA,OAAgB,S;MAAW,mB;QAAA,MAAe,S;MAAW,wB;QAAA,WaqlBR,M;MAp1B3E,QAAQ,  
E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,  
C;K;6GAwCX,yD;MAE0C,qB;QAAA,QAAiB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MA  
AO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb  
,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB  
,Q;MACHb,OAAO,C;K;yGAiCX,mC;MAEwC,qB;QAAA,QA2wByD,Q;MA3wBK,sB;QAAA,SA2wBL,Q;MA3  
wBoE,wB;QAAA,Wa4vBtF,M;MA3vB3E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,  
M;MACd,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;2FAYX,2B;MAEiC,mB;QAAA,MAuwb0C,Q;MAvwBJ,  
0B;QAAA,aAAsB,S;MACzF,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,YAAF,IAAkB,U;MACIB,O  
AAO,C;K;+GAYX,0B;MAE2C,uB;QAAA,UaqvBgC,Q;MArvBU,qB;QAAA,QAqvBV,Q;MApvBvE,QAAQ,E;  
MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;wEAgCX,4B;MAE+D,iBAAY,K  
AAZ,C;K;qFAyaY,qB;MAAQ,OAAU,S;K;6FAEd,qB;MAAQ,OAAc,a;K;uFAEzB,qB;MAAQ,OAAW,U;K;iFASx  
B,qB;MAAQ,OAAgE;K;iFAEX,qB;MAAQ,OAAQ,O;K;uFAEb,qB;MAAQ,OAAW,U;K;uFAS3B,qB;MAAQ,O  
AAW,U;K;mFAErB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;yFAShB,qB;MAAQ,OAAy,W;K;uF  
AerB,qB;MAAQ,OAAW,U;K;+FAEf,qB;MAAQ,OAAe,c;K;uFAE3B,qB;MAAQ,OAAW,U;K;uFAEnB,qB;MAA  
Q,OAAW,U;K;mFASrB,qB;MAAQ,OAAS,Q;K;iFAEiB,qB;MAAQ,OAAQ,O;K;6EAEiB,qB;MAAQ,OAAM,K;K  
;uFAET,qB;MAAQ,OAAW,U;K;qFASiB,qB;MAAQ,OAAU,S;K;qFAEiB,qB;MAAQ,OAAU,S;K;6EASr,qB;MA  
AQ,OAAM,K;K;mFAEX,qB;MAAQ,OAAS,Q;K;+EAEnB,qB;MAAQ,OAAO,M;K;+EAS/B,qB;MAAQ,OAAO,M  
;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;mFAShB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;  
MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;mFASd,qB;MAAQ,OAAQ,O;K;+EAEiB,qB;MAAQ,OAAM  
,K;K;+EAEB,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;6EASd,qB;  
MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;2FAEb,qB;MAAQ,OAAa,  
Y;K;6FAEpB,qB;MAAQ,OAAc,a;K;mFAE3B,qB;MAAQ,OAAS,Q;K;6EAS1B,qB;MAAQ,OAAM,K;K;6EAEd,q  
B;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;+EASjB,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAA  
S,Q;K;+EASrB,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;iFASjB,qB;MAAQ,OAAO,M;K;6FAER,q  
B;MAAQ,OAAc,a;K;qFAE1B,qB;MAAQ,OAAU,S;K;iFASb,qB;MAAQ,OAAO,M;K;uFAEZ,qB;MAAQ,OAAU,  
S;K;yFAS9B,qB;MAAQ,OAAy,W;K;+EAE1B,qB;MAAQ,OAAM,K;K;qFAEX,qB;MAAQ,OAAS,Q;K;iFAEnB,  
qB;MAAQ,OAAO,M;K;+EASrB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc,a;K;qFAS1B,qB;MAAQ,OA  
AU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;+EASX,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAS,Q;K;iFASn  
B,qB;MAAQ,OAAO,M;K;qFAEZ,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;kFASJ,qB;MAAQ,O  
AAQ,O;K;oFAEf,qB;MAAQ,OAAS,Q;K;8EAEpB,qB;MAAQ,OAAM,K;K;oFAEV,qB;MAAQ,OAAU,S;K;mFAS  
zC,qB;MAAQ,OAAS,Q;K;mFAEjB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;qFAEiB,qB;MAAQ,  
OAAU,S;K;wIEx+M7E,wM;MAEiD,qB;QAAA,QAAkB,I;MAAM,sB;QAAA,SAAmB,I;MAAM,2B;QAAA,cAA  
wB,I;MAAM,yB;QAAA,YAAsB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,sB;QAAA,  
SAAmB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,+B;



QAAA,kBAA4B,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,uB;QAAA,UAAoB,I;MAAM,4B;QAAA,eAAyB,I;MAAM,wB;QAAA,WAAqB,I;MAAM,uB;QAAA,UAAoB,I;MACrf,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wHAsDX,wM;MAEyC,qB;QAAA,QAAqB,S;MAAW,sB;QAAA,SAAsB,S;MAAW,2B;QAAA,cAA4B,S;MAAW,yB;QAAA,YAA0B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,sB;QAAA,SAAuB,S;MAAW,0B;QAAA,aAA0B,S;MAAW,0B;QAAA,aAA0B,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,+B;QAAA,kBAAmC,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,uB;QAAA,UAAwB,S;MAAW,4B;QAAA,eAA4B,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACtnB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;sHAYX,kN;MAEwC,wB;QAAA,WAA4C,S;MAAW,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MAC9IB,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;0HAsDX,wM;MAEOC,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACziB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;gHAYDX,wM;MAEqC,qB;QAAA,QAAc,S;MAAW,sB;QAAA,SAAc,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAmB,S;MAAW,0B;QAAA,aAAmB,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,+B;QAAA,kBAA4B,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACxB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;8HAqBX,gD;MAEsE,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;sIAoBX,gD;MAEgD,qB;QAAA,QAAiB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MA

AO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wHAWCX,wB;MAEyC,qB;QAAA,QAAiB,K;MAAO,qB;QAAA,QAAiB,K;MAC9E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;kGAYBX,oB;MAE8B,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MACnE,QA AQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;oHAYX,kC;MAEuC,q B;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MA CpI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF,IAAW,G;MACX ,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gGAYX,oB;MAE6B,mB;QAAA,MAAY,S;MAAW,mB;QAAA,MA AY,S;MAC5D,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;kHA YX,kC;MAEsC,qB;QAAA,QAAc,S;MAAW,qB;QAAA,QAAc,S;MAAW,mB;QAAA,MAAY,S;MAAW,mB;QAA A,MAAY,S;MACvH,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF, IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gIAeX,wB;MAE6C,qB;QAAA,QAAkB,S;MAAW,q B;QAAA,QAAkB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAA O,C;K;oIAeX,wB;MAE+C,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MACxF,QAAQ,E;MACR,EAAE,O AAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;4FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;o FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;8FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;kGASX,oB;MAE8 B,wB;QAAA,WAAkC,S;MAC5D,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;4FAUmE,qB;M AAQ,OAAO,M;K;8FAEd,qB;MAAQ,OAAQ,O;K;4FASrB,qB;MAAQ,OAAO,M;K;0GAER,qB;MAAQ,OAAc,a; K;8FAE7B,qB;MAAQ,OAAO,M;K;gGAEd,qB;MAAQ,OAAQ,O;K;8FASjB,qB;MAAQ,OAAO,M;K;gHAEL,qB; MAAQ,OAAiB,gB;K;wGASrC,qB;MAAQ,OAAa,Y;K;0GAEPB,qB;MAAQ,OAAc,a;K;wGAEvB,qB;MAAQ,OA Aa,Y;K;oFCroB7F,4B;MAE6E,iBAAY,KAAZ,C;K;iGASnB,qB;MAAQ,OAAS,Q;K;6FAEnB,qB;MAAQ,OAAO, M;K;+FAEd,qB;MAAQ,OAAQ,O;K;iGASF,qB;MAAQ,OAAU,S;K;-FAEnB,qB;MAAQ,OAAS,Q;K;mGAS3B,q B;MAAQ,OAAW,U;K;mGAEnB,qB;MAAQ,OAAW,U;K;6GC1D/E,mb;MAEmC,yB;QAAA,YAAkB,C;MAAG,q B;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,wB;QAAA,WAAmB,G;MAAI,kC;QAAA,qBAA6B,G; MAAI,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,2B;QAAA,cAAuB,E; MAAI,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UA AgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAA A,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO, sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B ,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA ,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAA O,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SAAc,C;MAAG,uB;QAAA,UAAoB,K; MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACl/B,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S; MACjB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,oB AAF,IAA0B,kB;MAC1B,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MA Cb,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF, IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,S AAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf, EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAA wB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;M ACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE ,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAA Y,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,U AAF,IAAgB,Q;MACHb,OAAO,C;K;6GC1BX,0C;MAEWc,oB;QAAA,OAAiB,I;MAAM,sB;QAAA,SAAmB,K;M AAO,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MACpI,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MAC Z,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;4EAmIX, 4B;MAEke,iBAAY,KAAZ,C;K;4EAEIE,qC;MAE2E,UAAAY,KAAZ,IAAqB,O;K;4EAIhG,4B;MAEuE,iBAAY,K AAZ,C;K;4EAEvE,qC;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAIpG,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,q

C;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAiGpG,4B;MAEoE,iBAAY,KAAZ,C;K;2EAEpE,qC;MAE4E,UAAAY,KAAZ,IAAqB,O;K;4EAKcJG,4B;MAE6E,iBAAY,KAAZ,C;K;4EAE7E,qC;MAEqF,UAAAY,KAAZ,IAAqB,O;K;4EAgP1G,4B;MAEqE,iBAAY,KAAZ,C;K;4EAERe,qC;MAE6E,UAAAY,KAAZ,IAAqB,O;K;uFJ57BIG,+H;MAE8B,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,8B;QAAA,iBAA0B,S;MAAW,oB;QAAA,OAAqB,S;MAAW,2B;QAAA,cAAmC,S;MAAW,qB;QAAA,QAAuB,S;MAAW,wB;QAAA,WAA6B,S;MAAW,yB;QAAA,YAAqB,S;MAAW,yB;QAAA,YAAsB,S;MAAW,wB;QAAA,WAAe,S;MAC5Z,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,OAAF,IAAa,K;MACb,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,WAAF,IAAiB,S;MACjB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,QAAF,IAAc,Q;MACd,OAAO,C;K;yFA0CX,uC;MAE+B,sB;QAAA,SAAiB,G;MAAK,0B;QAAA,aAAsB,I;MAAM,uB;QAAA,UAAmB,S;MACHG,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;qFAUgD,qB;MAAQ,OAAQ,E;K;mFAEX,qB;MAAQ,OAAQ,O;K;iFAEjB,qB;MAAQ,OAAO,M;K;mFAEd,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;qFASF,qB;MAAQ,OAAG,E;K;yFAER,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;qFAEd,qB;MAAQ,OAAQ,O;K;yFAEb,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qFAEjB,qB;MAAQ,OAAS,Q;K;uFAEjB,qB;MAAQ,OAAS,Q;K;mGAEV,qB;MAAQ,OAAgB,e;K;iGAEzB,qB;MAAQ,OAAe,c;K;qFAE9B,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;yFASzB,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;iFAErB,qB;MAAQ,OAAO,M;K;iFASD,qB;MAAQ,OAAO,M;K;iGAER,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAAU,S;K;yFAEjB,qB;MAAQ,OAAW,U;K;qFAErB,qB;MAAQ,OAAS,Q;K;yFAEf,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;qGAEnB,qB;MAAQ,OAAiB,gB;K;qFAS3B,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ,OAAQ,O;K;uFAEf,qB;MAAQ,OAAS,Q;K;mFASxB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;yFAEZ,qB;MAAQ,OAAU,S;K;qFAEpB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qGAET,qB;MAAQ,OAAiB,gB;K;+FKnR/F,gB;MAEKc,oB;QAAA,OAAgB,E;MAC9C,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;+FAiBX,8B;MAEKc,4B;QAAA,eAAqB,S;MAAW,oB;QAAA,OAAgB,E;MAC9E,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;0EAUX,4B;MAE6D,iBAAY,KAAZ,C;K;+GC6B7D,sJ;MAEsC,mB;QAAA,MA4GuD,M;MA5GG,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,E;MAAI,mB;QAAA,MAAe,E;MAAI,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,sB;QAAA,SAAmB,K;MAAO,oB;QAAA,OAAa,I;MAAM,uB;QAAA,UAAc,E;MAC/gB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,oBAAF,IAA0B,kB;MAClB,EAAE,QAAF,IAAc,M;MACd,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAWX,+B;MAEsE,oB;QAAA,OAAgB,S;MAClF,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qHASX,e;MAEyC,mB;QAAA,MAAe,E;MACpD,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;mHAyBX,+D;MAEqE,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACrK,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;iGAUwE,qB;MAAQ,OAAU,S;K;6FAEnB,qB;MAAQ,OAAS,Q;K;+FAEhB,qB;MAAQ,OAAU,S;K;2FASvB,qB;MAAQ,OAAO,M;K;yFAEhB,qB;MAAQ,OAAM,K;K;yFAEd,qB;MAAQ,OAAM,K;K;yGCrJ3F,uB;MAEsC,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,ORy9MW,S;MQx9MzE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;6HAuCX,mF;MAEGd,oB;QAAA,OAAa,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAA2C,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;QAAA,UA

AoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/S,QAAQ,E;MACR,EAAE,MAAF,IAA  
Y,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,  
OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MA  
ChB,OAAO,C;K;uGA2DX,qC;MAEqC,mC;QAAA,sBAAGC,K;MAAO,oB;QAAA,OA4UD,Q;MA3UvE,QAAQ,  
E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;yGAmBX,yC;MAEsC,u  
B;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACH,QAAQ,E;MACR,EA  
AE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;yGAsB  
X,2B;MAGI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FA8BX,s  
E;MAEoD,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAqB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAA  
A,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpL,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF  
,IAAgB,Q;MACHB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACI  
B,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GAuBX,0D;MAE2D,sB;QAAA,SAAkB,M;MAAQ,uB;QAAA,U  
AAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/J,QAAQ,E;MACR,EAAE,SAAF,IA  
Ae,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,  
UAAF,IAAgB,Q;MACHB,OAAO,C;K;2GAaX,qC;MAE4D,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAA0B,S;M  
ACf,OAAO,C;K;uHAuCX,mF;MAE6C,oB;QAAA,OAAa,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cA  
AuB,S;MAAW,sB;QAAA,SAAmD,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;Q  
AAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpT,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,Q  
AAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MA  
Cb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;q  
GA+BX,6D;MAEoC,4B;QAAA,eAAyB,K;MAAO,4B;QAAA,eAAyB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,yB  
;QAAA,YAAqB,S;MACnJ,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,cAAF,IAAoB,Y;MACpB,E  
AAE,YAAF,IAAkB,U;MACIB,EAAE,WAAF,IAAiB,S;MACjB,OAAO,C;K;yGakBX,4C;MAEsC,oB;QAAA,OA  
AgB,S;MAAW,uB;QAAA,UAAoB,S;MAAW,wB;QAAA,WAAsB,S;MAAW,uB;QAAA,UAA8B,S;MAC3J,QAA  
Q,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,  
SAAF,IAAe,O;MACf,OAAO,C;K;+FAkCmE,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;+FAEnB,q  
B;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;6FAStC,qB;MAAQ,OA  
AY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;uFAEvB,qB;MAAQ,OAAS,Q;K;qFAEnB,qB;MAAQ,OAAO,M;K;uF  
ASX,qB;MAAQ,OAAS,Q;K;yFAEjB,qB;MAAQ,OAAS,Q;K;qGAEX,qB;MAAQ,OAAe,c;K;iFAEhC,qB;MAAQ,  
OAAO,M;K;K;gCharE,0E;MAEoC,gC;QAAA,mBAA6B,K;MAAO,sB;QAAA,SAAkB,C;MAAG,qB;QAAA,QAAi  
B,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC3L,QAAQ,  
E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE  
,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;mFAU8E,q  
B;MAAQ,OAAG,E;K;+FAEL,qB;MAAQ,OAac,a;K;iFAE7B,qB;MAAQ,OAAO,M;K;yFAEX,qB;MAAQ,OAAW  
,U;K;+EAEvB,qB;MAAQ,OAAO,M;K;+EAEf,qB;MAAQ,OAAO,M;K;oEnIjIvG,yB;MAAA,kF;MAAA,0B;MAA  
A,uB;QAaI,IAAI,OAAO,CAAP,IAA8B,OAAO,KAAZC,C;UACI,MAAM,8BAAyB,wBAAqB,IAA9C,C;QAEV,O  
AAy,OAAL,IAAK,C;O;KAhBhB,C;0EAyCiC,qB;MAAQ,OAAA,SAAK,I;K;IoI7C9C,iC;K;;ICMA,4B;K;;IA6BA  
,gD;K;;IC5BA,qC;K;;IAyBA,+B;K;;IC6DqC,uC;MACjC,uB;QAAA,UAAsB,E;MACTB,qB;QAAA,+C;MADA,sB;  
MACA,kB;K;IAEA,4C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAKI,4E;MAGA,wE;K;;IAH  
A,mD;MAAA,gC;MAAA,2C;K;;IAGA,iD;MAAA,gC;MAAA,yC;K;;IARJ,sC;MAAA,2F;K;;IAAA,2C;MAAA,a;  
AAA,S;UAAA,gD;aAAA,O;UAAA,8C;;UAAA,+D;;K;;IA+ByB,4B;MACzB,8B;K;;IAGJ,qC;K;;IAyC6C,4C;MA  
CzC,8B;K;;ICpKqC,sC;MACrC,8B;K;;ICD4C,8B;K;kDAI5C,mB;MAA6D,c;;QIJisD7C,Q;QADhB,IAAI,mCAAs  
B,cAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,sB;QAAhB,OAAgB,cAAhB,C;UAAgB,2B;UAAM,IkJjsD6C,OIJi  
sD/B,SkJjsD+B,UJisD7C,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MkJIsDsD,iB;K;uDAE7D,oB;MACa,  
c;;QIJyqDG,Q;QADhB,IAAI,ckJxqDA,QIJwqDA,iBkJxqDA,QIJwqDsB,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QA  
CrB,OkJzqDZ,QIJyqDY,W;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CkJzqDP,oBIJyqDkB,OkJzqDIB  
,CIJyqDG,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MkJ1qDH,iB;K;2CAEJ,Y;MAAkC,qBAAQ,C;K;IAEq

B,qE;MAAA,qB;QAC3D,OAAI,OAAO,uBAAX,GAAiB,mBAAjB,GAA6C,SAAH,EAAG,C;O;K;4CADjD,Y;MAAkC,4BAAa,IAAb,EAAMb,GAAAnB,EAAwB,GAAXb,kBAA6B,wCAA7B,C;K;2CAIIC,Y;MAI4C,uBAAgB,IAAhB,C;K;mDAE5C,iB;MAI4D,yBAAgB,IAAhB,EAAsB,KAAtB,C;K;;IC/BhE,8B;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,4B;MAAA,+B;O;MACI,4C;MACA,kD;MACA,0C;MACA,8C;K;;IAHA,mC;MAAA,kB;MAAA,2B;K;;IACA,sC;MAAA,kB;MAAA,8B;K;;IACA,kC;MAAA,kB;MAAA,0B;K;;IACA,oC;MAAA,kB;MAAA,4B;K;;IAJJ,wB;MAAA,sH;K;;IAAA,6B;MAAA,a;aAAA,O;UAAA,gC;aAAA,U;UAAA,mC;aAAA,M;UAAA,+B;aAAA,Q;UAAA,iC;;UAAA,6D;;K;;IAOA,4B;MAKI,mD;MACA,2BAA4B,I;K;yCAE5B,Y;MAEiB,IAAN,I;MzJUX,IAAI,EyJXQ,mDzJWR,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;MyJZC,QAAM,oBAAN,M;aACH,M;UAAc,Y;UAAAd,K;aACA,O;UAAe,W;UAAf,K;;UACQ,wC;UAHL,K;;MAAP,W;K;sCAOJ,Y;MAIW,Q;MAHP,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,mD;MAEA,OAAO,2F;K;4DAGX,Y;MACI,iD;MACA,kB;MACA,OAAO,kD;K;+CAeX,iB;MAIL,2BAAy,K;MACZ,gD;K;sCAGJ,Y;MAII,+C;K;;ICjDkC,wB;MAoFtC,oC;MApFgE,6B;K;sCAIhE,Y;MAAuC,0C;K;2CAEvC,mB;MAAwD,uB;;QpJoU3C,Q;QADb,YAAy,C;QACC,sB;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IoJrUmE,OpJqUrD,IoJrUqD,UpJqUnE,C;YACI,sBAAO,K;YAAP,wB;;UACJ,qB;;QAEJ,sBAAO,E;;;MoJzUiD,0B;K;+CAExD,mB;MAA4D,sB;;QpJ6V5D,eAAoB,0BAAa,SAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI,IoJ/VsE,OpJ+VxD,QAAS,WoJ/V+C,UpJ+VtE,C;YACI,qBAAO,QAAS,Y;YAAhB,uB;;QAGR,qBAAO,E;;;MoJnWqD,yB;K;0CAE5D,Y;MAA+C,+CAAiB,CAAjB,C;K;kDAE/C,iB;MAAyD,+CAAiB,KAAjB,C;K;6CAEzD,8B;MAA8D,gCAAQ,IAAR,EAAC,SAAd,EAAYB,OAAzB,C;K;IAEIC,wD;MAAGf,uB;MAA/E,kB;MAAmC,4B;MAC5D,eAAyB,C;MAGrB,+DAAkB,gBAAIB,EAA6B,OAA7B,EAAsC,WAAK,KA3C,C;MACA,eAAa,UAAU,gBAAV,I;K;iDAGjB,iB;MACI,+DAAkB,KAAIB,EAAYB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAy,KAAZ,IAAL,C;K;4FAGY,Y;MAAQ,mB;K;;oCAGnC,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,2DAAC,IAAd,EAAoB,KAApB,C;K;sCAGX,Y;MAG+B,oEAAgB,IAAhB,C;K;IAE/B,2C;MAAA,oB;MACI,eACsB,C;K;kDAEtB,Y;MAAkC,sBAAQ,gB;K;+CAEIC,Y;MAEe,gB;MADX,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACX,iE;MAAX,OAAO,+B;K;;IAO0B,sD;MAHzC,oB;MAGwD,iD;MAGhD,gEAAMb,KAAAnB,EAA0B,WAAkB,KAA5C,C;MACA,eAAa,K;K;0DAGjB,Y;MAAsC,sBAAQ,C;K;wDAE9C,Y;MAAgC,mB;K;uDAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B;MAC1B,OAAO,yBAAL,mCAAJ,EAAL,YAAJ,E;K;4DAGX,Y;MAAoC,sBAAQ,CAAR,I;K;;IAGxC,kC;MAAA,sC;K;iEACI,uB;MACI,IAAI,QAAQ,CAAR,IAAa,SAAS,IAA1B,C;QACI,MAAM,8BAA0B,YAAS,KAAT,gBAAuB,IAAjD,C;;K;kEAIId,uB;MACI,IAAI,QAAQ,CAAR,IAAa,QAAQ,IAAzB,C;QACI,MAAM,8BAA0B,YAAS,KAAT,gBAAuB,IAAjD,C;;K;iEAIId,oC;MACI,IAAI,YAAy,CAAZ,IAAiB,UAAU,IAA/B,C;QACI,MAAM,8BAA0B,gBAAa,SAAb,mBAAkC,OAAIC,gBAAkD,IAA5E,C;;MAEV,IAAI,YAAy,OAAhB,C;QACI,MAAM,gCAAYB,gBAAa,SAAb,oBAAMC,OAA5D,C;;K;kEAIId,sC;MACI,IAAI,aAAa,CAAb,IAAkB,WAAW,IAAjC,C;QACI,MAAM,8BAA0B,iBAAC,UAAAd,oBAAqC,QAArC,gBAAsD,IAAhF,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAAYB,iBAAC,UAAAd,qBAAsC,QAA/D,C;;K;+DAId,a;MAEc,UACsB,M;MAFhC,iBAAE,C;MACL,mB;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,aAAW,MAAK,UAAAL,SAAiB,6DAAiB,CAALC,K;;MAEf,OAAO,U;K;6DAGX,oB;MAIiB,Q;MAHb,IAAI,CAAE,KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MAEjC,oBAAoB,KAAm,W;MACb,mB;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,gBAAgB,aAAc,O;QAC9B,IAAI,cAAQ,SAAR,CAAJ,C;UACI,OAAO,K;;MAGf,OAAO,I;K;;IAjDf,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;ICnFwC,uB;MAyHxC,mC;MAzCA,uBAC6B,I;MAMc7B,yBACsC,I;K;8CAnHtC,e;MACI,OAAO,6BAAC,GAAd,S;K;gDAGX,iB;MAAwE,gBAAR,Y;MAAQ,c;;QrJwrDxD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IqJxrDwD,OrJwrD1C,OqJxrD6C,MAAH,QrJwrDxD,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MqJzrDyD,iB;K;kDAEhE,iB;MAEI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,UAAU,KAAM,I;MACHb,YAAy,KAAM,M;MpKmNO,Q;MoKINzB,epKkN4C,CAAnB,mDAAMb,YoKINzB,GpKkNyB,C;MoKhN5C,IAAI,eAAS,QAAT,CAAJ,C;QACI,OAAO,K;;MAIP,6B;MAAA,W;QpK4NqB,U;QoK5ND,UpK4NoB,CAAnB,uDAAMb,oBoK5NP,GpK4NO,C;;MoK5N5C,W;QACI,OAAO,K;;MAGX,OAAO,I;K;mCAIX,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAyB,OAAO,K;MACHc,IAAI,cAAQ,KAAM,KAAIB,C;QAAwB,OAAO,K;MAEV,gBAAd,KAAM,Q;MAAQ,c;;QrJmoDT,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CqJnoDK,2BrJmoDM,OqJnoDN,CrJmoDT,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MqJpoDH,iB;K;sCAGJ,e

;MAAwC,Q;MAAA,4CAAc,GAAd,8B;K;qCAGxC,Y;MAK+B,OAAQ,SAAR,YAAQ,C;K;oCAEvC,Y;MAAkC,q  
BAAQ,C;K;mFACnB,Y;MAAQ,OAAA,YAAQ,K;K;IAWnB,0E;MAAA,wC;MAAS,sB;K;8EACb,mB;MAAsD,+  
CAAY,OAAZ,C;K;IAI3C,sG;MAAA,kD;K;8FACH,Y;MAAkC,OAAA,0BAAc,U;K;2FACHD,Y;MAAyB,OAAA,  
0BAAc,OAAO,I;K;;wEAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,+F;K;sHAMmB,Y;MAAQ,OAAA,qBAAiB,  
K;K;;mFAB5D,Y;MACI,IAAI,4BAAJ,C;QACI,+E;;MacJ,OAAO,mC;K;IAOwD,uD;MAAA,qB;QAAE,2CAAS,E  
AAT,C;O;K;qCAAzE,Y;MAAkC,OAAQ,eAAR,YAAQ,EAAa,IAAb,EAAMb,GAAnB,EAAwB,GAAxB,kBAA6B  
,iCAA7B,C;K;+CAE1C,iB;MAAuD,+BAAS,KAAM,IAAf,IAAsB,GAAtB,GAA4B,wBAAS,KAAM,MAAf,C;K;+  
CAEnF,a;MAAwC,OAAI,MAAM,IAAV,GAAGB,YAAhB,GAAoC,SAAF,CAAE,C;K;IAWtD,4E;MAAA,wC;MA  
AS,6B;K;gFACf,mB;MAAsE,iDAAc,OAAc,C;K;IAI3D,wG;MAAA,kD;K;gGACH,Y;MAAkC,OAAA,0BAAc,U;  
K;6FACHD,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;;0EAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,iG;K;wHA  
MmB,Y;MAAQ,OAAA,qBAAiB,K;K;;qFAB5D,Y;MACI,IAAI,8BAAJ,C;QACI,mF;;MacJ,OAAO,qC;K;oDAMf,  
e;MAA8D,gBAAR,Y;MAAQ,sB;;QrJmJ9C,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAm,IqJnJsD  
,OrJmJxC,OqJnJ2C,IAAH,MrJmJtD,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MqJpJ+C,yB;K;IAEt  
D,iC;MAAA,qC;K;4DAEI,a;MAAiE,gC;MAAX,OAAU,CAAC,kBAAN,CAAM,0DAAMb,CAApB,KAA4B,oBA  
AjC,CAAiC,8DAAqB,CAAjD,C;K;4DACHe,a;MAAyD,OAAU,SAAL,CAAO,IAAF,mBAAL,CAAY,MAAP,C;K;  
0DACnE,oB;MACI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,OAAO,OAAA,CAAE,IAAF,EAAS,KAAM,IAAf,  
KAAsB,OAAA,CAAE,MAAF,EAAW,KAAM,MAAjB,C;K;;;IANrC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;IC  
hIqC,uB;MAKBrC,mC;MAIB+D,6B;K;mCAE/D,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,  
0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,sDAAU,IAAV,EAAGB,KAAhB,C;K;qCAGX,Y;MAG+B,qEAakB,I  
AAIB,C;K;IAE/B,iC;MAAA,qC;K;gEACI,a;MAEoB,Q;MADhB,iBAAE,C;MACC,mB;MAAhB,OAAgB,cAAhB,  
C;QAAgB,yB;QACC,U;QAAb,2BAAa,yEAAuB,CAApC,K;;MAEJ,OAAO,U;K;wDAGX,oB;MACI,IAAI,CAAE,  
KAAF,KAAU,KAAM,KAAPB,C;QAA0B,OAAO,K;MACjC,OAAO,CtK8OsG,qBsK9Of,KtK8OwF,C;K;;;IsKzP  
rH,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;;MCghBA,kC;MA9hBA,cAAwB,C;MACxB,yB;MAEA,sBAAyB,C;;  
kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;4CA8BA,uB;MAOI,IAAI,cAAc,CAAlB,C;QAAqB,MAAM,6B  
AAsB,mBAAtB,C;MAC3B,IAAI,eAAe,kBAAY,OAA/B,C;QAAqC,M;MACrC,IAAI,uBAAgB,qDAAPB,C;QACI,  
qBAAc,gBAAyB,gBAAZ,WAAy,EAAC,EAAd,CAAZB,O;QACd,M;;MAGJ,kBAakB,uDAAY,kBAAy,OAAxB,  
EAA8B,WAA9B,C;MACIB,oBAAa,WAAb,C;K;0CAGJ,uB;MAII,kBAakB,gBAAMb,WAAAnB,O;M5J20BtU,4J  
10BI,kB5J00BJ,E4J10ByB,W5J00BzB,E4J10BsC,C5J00BtC,E4J10ByC,W5J00BzC,E4J10B+C,kBAAy,O5J00B3  
D,C;MAAA,U4Jz0BI,kB5Jy0BJ,E4Jz0ByB,W5Jy0BzB,E4Jz0BsC,kBAAy,OAAZ,GAAMb,WAAAnB,I5Jy0BtC,E4  
Jz0B+D,C5Jy0B/D,E4Jz0BkE,W5Jy0BIE,C;M4Jx0BI,cAAO,C;MACP,qBAAc,W;K;yCAGIB,yB;MAGW,Q;MAA  
P,OAAO,2BAAY,aAAZ,4D;K;yCAGX,iB;MAA2C,OAAI,SAAS,kBAAY,OAAZB,GAA+B,QAAQ,kBAAY,OAA  
pB,IAA/B,GAA6D,K;K;yCAExG,iB;MAA2C,OAAI,QAAQ,CAAZ,GAAe,QAAQ,kBAAY,OAApB,IAAf,GAA6C  
,K;K;2CAExF,iB;MACoD,0BAAY,cAAO,KAAP,IAAZ,C;K;yCAEpD,iB;MAA2C,OAAI,UAAqB,cAAZ,kBAAY,  
CAAzB,GAAoC,CAAPC,GAA2C,QAAQ,CAAR,I;K;yCAEtF,iB;MAA2C,OAAI,UAAAS,CAAb,GAA4B,cAAZ,kB  
AAy,CAA5B,GAA2C,QAAQ,CAAR,I;K;mCAEtF,Y;MAAkC,qBAAQ,C;K;iCAE1C,Y;MAGwB,IAAI,cAAJ,C;Q  
AAe,MAAM,2BAAuB,sBAAvB,C;;QAnBIC,Q;QAmBa,OAnBb,2BAmbkG,WAnBIG,4D;;K;uCAqBX,Y;MAG+B  
,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAe,I;;QAxBnC,U;QAwBoB,OAxBpB,6BAwByD,WAxBzD,gE;;MAwBoB,W  
;K;gCAE/B,Y;MAGuB,IAAI,cAAJ,C;QAe,MAAM,2BAAuB,sBAAvB,C;;QA7BjC,Q;QA6BY,OA7BZ,2BAQyC  
,mBAAy,cAqB0D,sBArB1D,IAAZ,CARzC,4D;;K;sCA+BX,Y;MAG8B,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAe,I;;  
QAICIC,U;QAKcMB,OAlCnB,6BAQyC,mBAAy,cA0BiB,sBA1BjB,IAAZ,CARzC,gE;;MAKcMB,W;K;0CAE9B,  
mB;MAII,sBAAE,YAAO,CAAP,IAAf,C;MAEA,cAAO,mBAAy,WAAZ,C;MACP,mBAAy,WAAZ,IAAoB,O;M  
ACpB,wBAAQ,CAAR,I;K;yCAGJ,mB;MAII,sBAAE,YAAO,CAAP,IAAf,C;MAEA,mBA7CgD,mBAAy,cA6CIC,  
SA7CkC,IAAZ,CA6ChD,IAAMc,O;MACnC,wBAAQ,CAAR,I;K;uCAGJ,Y;MAII,IAAI,cAAJ,C;QAe,MAAM,2  
BAAuB,sBAAvB,C;MA7Dd,Q;MA+DP,cA/DO,2BA+DmB,WA/DnB,4D;MAGeP,mBAAy,WAAZ,IAAoB,I;MA  
CpB,cAAO,mBAAy,WAAZ,C;MACP,wBAAQ,CAAR,I;MACA,OAAO,O;K;6CAGX,Y;MAGqC,OAAI,cAAJ,G  
AAe,IAAf,GAAyB,kB;K;sCAE9D,Y;MAII,IAAI,cAAJ,C;QAe,MAAM,2BAAuB,sBAAvB,C;MAErB,wBAzEg  
D,mBAAy,cAyEtB,sBAzEsB,IAAZ,C;MARzC,Q;MAkFP,cAlFO,2BAkFmB,iBAIFnB,4D;MAmFP,mBAAy,iBA  
AZ,IAAiC,I;MACjC,wBAAQ,CAAR,I;MACA,OAAO,O;K;4CAGX,Y;MAGoC,OAAI,cAAJ,GAAe,IAAf,GAAyB

,iB;K;qCAE7D,mB;MAEI,mBAAQ,OAAR,C;MACA,OAAO,I;K;uCAGX,0B;MACI,oCAAa,4BAAmB,KAAAnB,EAA0B,SAA1B,C;MAEb,IAAI,UAAS,SAAb,C;QACI,mBAAQ,OAAR,C;QACA,M;;aACG,IAAI,UAAS,CAAb,C;QACH,oBAAS,OAAT,C;QACA,M;;MAGJ,sBA Ae, YAAO,CAAP,IAAf,C;MA2BA,oBAjIgd,mBAAY,cAiI1B,KAJI0B,IAAZ,C;MAMhD,IAAI,QAAS,SAAD,GAAQ,CAAR,IAAe,CAA3B,C;QAEI,+BAA+B,mBAAY,aAAZ,C;QAC/B,sBAAsB,mBAAY,WAAZ,C;QAEtB,IAAI,4BAA4B,WAAhC,C;UACI,mBAAY,eAAZ,IAA+B,mBAAY,WAAZ,C;U5JgrB3C,U4J/qBY,kB5J+qBZ,E4J/qBiC,kB5J+qBjC,E4J/qB8C,W5J+qB9C,E4J/qBoD,cAAO,CAAP,I5J+qBpD,E4J/qB8D,2BAA2B,CAA3B,I5J+qB9D,C;;UAAA,U4J7qBY,kB5J6qBZ,E4J7qBiC,kB5J6qBjC,E4J7qB8C,cAAO,CAAP,I5J6qB9C,E4J7qBwD,W5J6qBxD,E4J7qB8D,kBAAY,O5J6qB1E,C;U4J5qBY,mBAAY,kBAAY,OAAZ,GAAMB,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U5J4qBhD,U4J3qBY,kB5J2qBZ,E4J3qBiC,kB5J2qBjC,E4J3qB8C,C5J2qB9C,E4J3qBiD,C5J2qBjD,E4J3qBoD,2BAA2B,CAA3B,I5J2qBpD,C;;Q4JxqBQ,mBAAY,wBAAZ,IAAwC,O;QACxC,cAAO,e;;QAGP,WArJ4C,mBAAY,cAqJ/B,SArJ+B,IAAZ,C;QAUJ5C,IAAI,gBAAGB,IAApB,C;U5JkqBR,U4JjqBY,kB5JiqBZ,E4JjqBiC,kB5JiqBjC,E4JjqB8C,gBAAGB,CAAhB,I5JiqB9C,E4JjqBiE,a5JiqBjE,E4JjqBgF,I5JiqBhF,C;;UAAA,U4J/pBY,kB5J+pBZ,E4J/pBiC,kB5J+pBjC,E4J/pB8C,C5J+pB9C,E4J/pBiD,C5J+pBjD,E4J/pBoD,I5J+pBpD,C;U4J9pBY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAMB,CAAnB,IAAZ,C;U5J8pB7B,U4J7pBY,kB5J6pBZ,E4J7pBiC,kB5J6pBjC,E4J7pB8C,gBAAGB,CAAhB,I5J6pB9C,E4J7pBiE,a5J6pBjE,E4J7pBgF,kBAAY,OAAZ,GAAMB,CAAnB,I5J6pBhF,C;;Q4J1pBQ,mBAAY,aAAZ,IAA6B,O;;MAEjC,wBAAQ,CAAR,I;K;oDAGJ,mC;MAGkD,UAIxB,M;MANtB,eAAe,QAAS,W;MAEsB,OAAZ,kBAAY,O;MAA9C,iBAAC,aAAd,wB;QACI,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,K;QACzB,mBAAY,KAAZ,IAAqB,QAAS,O;;MAEZ,oB;MAAtB,mBAAC,CAAd,8B;QACI,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,K;QACzB,mBAAY,OAAZ,IAAqB,QAAS,O;;MAGIC,wBAAQ,QAAS,KAAjB,I;K;0CAGJ,oB;MACI,IAAI,QAAS,UAAAb,C;QAaWB,OAAO,K;MAC/B,sBA Ae,IAAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MACA,8BAAtLgD,mBAAY,cAsLvB,SAtLuB,IAAZ,CAsLhD,EA A4C,QAA5C,C;MACA,OAAO,I;K;0CAGX,2B;MACI,oCAAa,4BAAmB,KAAAnB,EAA0B,SAA1B,C;MAEb,IAAI,QAAS,UAAAb,C;QACI,OAAO,K;aACJ,IAAI,UAAS,SAAb,C;QACH,OAAO,oBAAO,QAAP,C;;MAGX,sBA Ae,IAAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MAEA,WArMgD,mBAAY,cAqMnC,SArMmC,IAAZ,C;MAsMhD,oBAtMgD,mBAAY,cAsM1B,KAtM0B,IAAZ,C;MAuMhD,mBAAmB,QAAS,K;MAE5B,IAAI,QAAS,SAAD,GAAQ,CAAR,IAAe,CAA3B,C;QAGI,kBAAkB,cAAO,YAAP,I;QAEIb,IAAI,iBAAiB,WAArB,C;UACI,IAAI,eAAe,C AAnB,C;Y5J0mBZ,U4JzmBgB,kB5JymBhB,E4JzmBqC,kB5JymBrC,E4JzmBkD,W5JymBiD,E4JzmB+D,W5Jym B/D,E4JzmBqE,a5JymBrE,C;;Y4JvmBgB,4BA Ae,kBAAY,OAA3B,I;YACA,sBAAsB,gBAAGB,WAAhB,I;YACt B,kBAAkB,kBAAY,OAAZ,GAAMB,WAAAnB,I;YAEIb,IAAI,eAAe,eAAnB,C;c5JmmBhB,U4JlmBoB,kB5JkmBp B,E4JlmByC,kB5JkmBzC,E4JlmBsD,W5JkmBtD,E4JlmBmE,W5JkmBnE,E4JlmByE,a5JkmBzE,C;;cAAA,U4Jhm BoB,kB5JgmBpB,E4JhmByC,kB5JgmBzC,E4JhmBsD,W5JgmBtD,E4JhmBmE,W5JgmBnE,E4JhmByE,cAAO,W AAP,I5JgmBzE,C;cAAA,U4J/lBoB,kB5J+lBpB,E4J/lByC,kB5J+lBzC,E4J/lBsD,C5J+lBtD,E4J/lByD,cAAO,WAA P,I5J+lBzD,E4J/lB6E,a5J+lB7E,C;;;UAAA,U4J3lBY,kB5J2lBZ,E4J3lBiC,kB5J2lBjC,E4J3lB8C,W5J2lB9C,E4J3l B2D,W5J2lB3D,E4J3lBiE,kBAAY,O5J2lB7E,C;U4J1lBY,IAAI,gBAAGB,aAApB,C;Y5J0lBZ,U4JzlBgB,kB5JylB hB,E4JzlBqC,kB5JylBrC,E4JzlBkD,kBAAY,OAAZ,GAAMB,YAAAnB,I5JylBiD,E4JzlBmF,C5JylBnF,E4JzlBsF,a5 JylBtF,C;;YAAA,U4JvlBgB,kB5JulBhB,E4JvlBqC,kB5JulBrC,E4JvlBkD,kBAAY,OAAZ,GAAMB,YAAAnB,I5Jul BiD,E4JvlBmF,C5JulBnF,E4JvlBsF,Y5JulBtF,C;YAAA,U4JtlBgB,kB5JslBhB,E4JtlBqC,kB5JslBrC,E4JtlBkD,C5J slBiD,E4JtlBqD,Y5JslBrD,E4JtlBmE,a5JslBnE,C;;Q4JnlBQ,cAAO,W;QACP,8BAAuB,mBAAY,gBAAGB,YAAh B,IAAZ,CAAvB,EAaKE,QAAIE,C;;QAIA,2BAA2B,gBAAGB,YAAhB,I;QAE3B,IAAI,gBAAGB,IAApB,C;UACI, IAAI,QAAO,YAAP,SAAuB,kBAAY,OAAvC,C;Y5J2kBZ,U4J1kBgB,kB5J0kBhB,E4J1kBqC,kB5J0kBrC,E4J1kB kD,oB5J0kBiD,E4J1kBwE,a5J0kBxE,E4J1kBuF,I5J0kBvF,C;;Y4JxkBgB,IAAI,wBAAwB,kBAAY,OAAxC,C;c5J wkBhB,U4JvkBoB,kB5JukBpB,E4JvkByC,kB5JukBzC,E4JvkBsD,uBAAuB,kBAAY,OAAAnC,I5JukBtD,E4JvkB+ F,a5JukB/F,E4JvkB8G,I5JukB9G,C;;c4JrkBoB,mBAAmB,OAAO,YAAP,GAAsB,kBAAY,OAAIC,I;c5JqkBvC,U4 JpkBoB,kB5JokBpB,E4JpkByC,kB5JokBzC,E4JpkBsD,C5JokBtD,E4JpkByD,OAAO,YAAP,I5JokBzD,E4JpkB8E ,I5JokB9E,C;cAAA,U4JnkBoB,kB5JmkBpB,E4JnkByC,kB5JmkBzC,E4JnkBsD,oB5JmkBtD,E4JnkB4E,a5JmkB5 E,E4JnkB2F,OAAO,YAAP,I5JmkB3F,C;;;UAAA,U4J/jBY,kB5J+jBZ,E4J/jBiC,kB5J+jBjC,E4J/jB8C,Y5J+jB9C, E4J/jB4D,C5J+jB5D,E4J/jB+D,I5J+jB/D,C;U4J9jBY,IAAI,wBAAwB,kBAAY,OAAxC,C;Y5J8jBZ,U4J7jBgB,kB 5J6jBhB,E4J7jBqC,kB5J6jBrC,E4J7jBkD,uBAAuB,kBAAY,OAAAnC,I5J6jBiD,E4J7jB2F,a5J6jB3F,E4J7jB0G,kB

AAy,O5J6jBtH,C;;YAAA,U4J3jBgB,kB5J2jBhB,E4J3jBqC,kB5J2jBrC,E4J3jBkD,C5J2jBID,E4J3jBqD,kBAAY,  
OAAZ,GAAMb,YAAnB,I5J2jBrD,E4J3jBsF,kBAAY,O5J2jBIG,C;;YAAA,U4J1jBgB,kB5J0jBhB,E4J1jBqC,kB5J  
0jBrC,E4J1jBkD,oB5J0jBID,E4J1jBwE,a5J0jBxE,E4J1jBuF,kBAAY,OAAZ,GAAMb,YAAnB,I5J0jBvF,C;;Q4Jvj  
BQ,8BAAuB,aAAvB,EAAsC,QAAAT,C;;MAGJ,OAAO,I;K;uCAGX,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAYB,  
SAAzB,C;MAjRN,Q;MAmRP,OAnRO,2BAQyC,mBAAY,cA2Q3B,KA3Q2B,IAAZ,CARzC,4D;K;uCArRX,0B;  
MACI,oCAAA,2BAAkB,KAAIB,EAAYB,SAAzB,C;MAEb,oBAjRgD,mBAAY,cAiR1B,KAjR0B,IAAZ,C;MARzC  
,Q;MA0RP,iBA1RO,2BA0RsB,aA1RtB,4D;MA2RP,mBAAY,aAAZ,IAA6B,O;MAE7B,OAAO,U;K;OCAGX,mB;  
MAAoD,0BAAQ,OAAr,MAAoB,E;K;yCAExE,mB;MAIsB,IAIA,IAJA,EAIuB,M;MAPzC,WA3RgD,mBAAY,c  
A2RnC,SA3RmC,IAAZ,C;MA6RhD,IAAI,cAAO,IAAX,C;QACI,iBAAC,WAAAd,UAAyB,IAAzB,U;UACI,IAAI,g  
BAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAAO,QAAQ,WAAr,I;;aAE3C,IAAI,eAAQ,IAAZ,C;QACW,  
kB;QAAuB,SAAZ,kBAAY,O;QAArC,qD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO  
,UAAQ,WAAr,I;;QAE9C,mBAAC,CAAd,YAAsB,IAAtB,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,  
C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;MAIID,OAAO,E;K;6CAGX,mB;MAIsC,UAOJ,  
MAPI,EA0a,M;MAV/C,WA9SgD,mBAAY,cA8SnC,SA9SmC,IAAZ,C;MAGThD,IAAI,cAAO,IAAX,C;QACKC,k  
B;QAA9B,iBAAC,OAAO,CAAP,IAAd,yB;UACI,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAA  
O,QAAQ,WAAr,I;;aAE3C,IAAI,cAAO,IAAX,C;QACH,mBAAC,OAAO,CAAP,IAAd,aAA8B,CAA9B,Y;UACI,I  
AAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;QA  
EpB,uBAAZ,kBAAY,C;QAAiB,oB;QAA3C,wD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,  
OAAO,UAAQ,WAAr,I;;MAIID,OAAO,E;K;wCAGX,mB;MACI,YAAY,mBAAQ,OAAr,C;MACZ,IAAI,UAAS  
,EAAb,C;QAAiB,OAAO,K;MACxB,sBAAS,KAAT,C;MACA,OAAO,I;K;4CAGX,iB;MACI,oCAAA,2BAAkB,K  
AAIB,EAAYB,SAAzB,C;MAEb,IAAI,UAAS,sBAAb,C;QACI,OAAO,iB;aACJ,IAAI,UAAS,CAAb,C;QACH,OA  
AO,kB;;MAGX,oBAhVgD,mBAAY,cAgV1B,KAhV0B,IAAZ,C;MARzC,Q;MAyVP,cAzVO,2BAyVmB,aAzVnB,  
4D;MA2VP,IAAI,QAAQ,aAAS,CAArB,C;QAEI,IAAI,iBAAiB,WAArB,C;U5JoeR,U4JneY,kB5JmeZ,E4JneiC,kB  
5JmejC,E4Jne8C,cAAO,CAAP,I5Jme9C,E4JnewD,W5JmexD,E4Jne8D,a5Jme9D,C;;UAAA,U4JjeY,kB5JieZ,E4Jj  
eiC,kB5JiejC,E4Jje8C,C5Jie9C,E4JjeiD,C5JiejD,E4JjeoD,a5JiepD,C;U4JheY,mBAAY,CAAZ,IAAiB,mBAAY,kB  
AAY,OAAZ,GAAMb,CAAnB,IAAZ,C;U5Jge7B,U4J/dY,kB5J+dZ,E4J/diC,kB5J+djC,E4J/d8C,cAAO,CAAP,I5J+  
d9C,E4J/dwD,W5J+dxD,E4J/d8D,kBAAY,OAAZ,GAAMb,CAAnB,I5J+d9D,C;;Q4J5dQ,mBAAY,WAAZ,IAAoB  
,I;QACpB,cAAO,mBAAY,WAAZ,C;;QAGP,wBAjW4C,mBAAY,cAiWIB,sBAjWkB,IAAZ,C;QAmW5C,IAAI,iB  
AAiB,iBAArB,C;U5JsdR,U4JrdY,kB5JqdZ,E4JrdiC,kB5JqjdC,E4Jrd8C,a5Jqd9C,E4Jrd6D,gBAAgB,CAAhB,I5Jq  
d7D,E4JrdgF,oBAAoB,CAApB,I5JqdhF,C;;UAAA,U4JndY,kB5JmdZ,E4JndiC,kB5JmdjC,E4Jnd8C,a5Jmd9C,E4J  
nd6D,gBAAgB,CAAhB,I5Jmd7D,E4JndgF,kBAAY,O5Jmd5F,C;U4JldY,mBAAY,kBAAY,OAAZ,GAAMb,CAA  
nB,IAAZ,IAAoC,mBAAY,CAAZ,C;U5JkdhD,U4JjdY,kB5JidZ,E4JjdiC,kB5JidjC,E4Jjd8C,C5Jid9C,E4JjdiD,C5Ji  
djD,E4JjdoD,oBAAoB,CAApB,I5JidpD,C;;Q4J9cQ,mBAAY,iBAAZ,IAAiC,I;;MAErC,wBAAQ,CAAR,I;MAEA,  
OAAO,O;K;6CAGX,oB;MAAkE,0B;;QAA5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB5KoxOnB,  
YAAQ,C;;Q4KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2  
XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAC,WAAAd,UAAyB,IAAzB,U;YACI,cAAc,m  
BAAY,KAAZ,C;YAGd,IAjBsE,CAAU,wBAiBIE,0EAjBkE,CAiBhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,Y  
AAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;UAGE,oB;U  
AAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,I  
A/BsE,CAAU,wBA+BIE,kFA/BkE,CA+BhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAA  
W,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAc,mBAAY,OAAZ,  
C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA5CsE,CAAU,wBA4CIE,kFA5CkE,CA4ChF,C;cACI,mBAAY,OAA  
Z,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,  
UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;MAvDuD,6B;K;6CAEIE,oB;MAAkE,0B;;QAW5C,wD;QART,aAAL,  
IAAK,U;QAAL,Y;UAA8B,SAAZ,kB5KoxOnB,YAAQ,C;;Q4KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA  
1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,  
iBAAC,WAAAd,UAAyB,IAAzB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAf+E,wBAejE,0EAfIE,CAe/E,C;cACI,  
mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OA



AX,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA7B+E,wBA6BJE,kFA7BiE,CA6B/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAc,CAAd,YAAsB,IAAtB,Y;YACI,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA1C+E,wBA0CjE,kFA1CiE,CA0C/E,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;MArDuD,6B;K;2CAEIE,qB;MASsB,IAII,IAJJ,EAKM,MALN,EAaA,MAbA,EAauB,MAbvB,EAKBI,MAIBJ,EAmBM,MAAnBN,EA+BI,M;MAvCb,aAAL,IAAK,U;MAL,Y;QAA8B,SAAZ,kB5KoxOnB,YAAQ,C;;M4KpxOX,W;QACI,OAAO,K;MAEX,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;MA2XhD,cAAc,W;MACd,eAAe,K;MAEf,IAAI,cAAO,IAAX,C;QACI,iBAAc,WAAAd,UAAyB,IAAzB,U;UACI,cAAc,mBAAY,KAAZ,C;UAGd,IAAI,UAAU,0EAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;YAEzB,WAAW,I;;QAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;QAGE,oB;QAAuB,SAAZ,kBAAY,O;QAArC,uD;UACI,gBAAc,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;YAEzB,WAAW,I;;QAGnB,UAAU,mBAAY,OAAZ,C;QAEV,mBAAc,CAAd,YAAsB,IAAtB,Y;UACI,gBAAc,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,OAAZ,IAAuB,S;YACvB,UAAU,mBAAY,OAAZ,C;;YAEV,WAAW,I;;;MAIvB,IAAI,QAAJ,C;QACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;MAEX,OAAO,Q;K;iCAGX,Y;MACI,WA7agD,mBAAY,cA6anC,SA7amC,IAAZ,C;MA8ahD,IAAI,cAAO,IAAX,C;QACgB,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAAX,EAAiB,IAAjB,C;;QACT,IvKpS6C,CAAC,cuKoS9C,C;UACS,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAAX,EAAiB,kBAAY,OAA7B,C;UACA,OAAZ,kBAAY,EAAK,IAAL,EAAW,CAAX,EAAC,IAAd,C;;MAEHb,cAAO,C;MACP,YAAO,C;K;2CAGX,iB;MAGe,IAAC,IAAD,EAAC,M;MAfP,WACW,eAAC,OAAI,KAAM,OAAN,IAAc,SAAlB,GAAwB,KAAxB,GAAmC,aAAa,KAAb,EAAoB,SAApB,CAApC,uB;MAEX,WA7bgD,mBAAY,cA6bnC,SA7bmC,IAAZ,C;MA8bhD,IAAI,cAAO,IAAX,C;Q5J2XJ,U4J1XQ,kB5J0XR,E4J1X6B,I5J0X7B,EAD+F,CAC/F,E4J1XgD,W5J0XhD,E4J1XiE,I5J0XjE,C;;Q4JzXW,IvKpT6C,CAAC,cuKoT9C,C;U5JyXX,U4JxXQ,kB5JwXR,E4JxX6B,I5JwX7B,E4JxXuD,C5JwXvD,E4JxXuE,W5JwXvE,E4JxXwF,kBAAY,O5JwXpG,C;UAAA,U4JvXQ,kB5JuXR,E4JvX6B,I5JuX7B,E4JvXuD,kBAAY,OAAZ,GAAmB,WAAAnB,I5JuXvD,E4JvX6F,C5JuX7F,E4JvX2G,I5JuX3G,C;;M4JrXI,IAAI,IAAK,OAAL,GAAy,SAAhB,C;QACI,KAAK,SAAL,IAAa,I;;MAIjB,OAAO,qD;K;mCAGX,Y;MAEI,OAAO,qBAAQ,gBAAmB,SAAnB,OAAR,C;K;+CAGX,iB;MAC0D,4BAAQ,KAAR,C;K;+CAC1D,Y;MAA0C,qB;K;IAE1C,gC;MAAA,oC;MACI,0BxHriBuC,E;MwHsiBvC,sBAAiC,U;MACjC,4BAAuC,E;K;yDAEvC,oC;MAEI,kBAAkB,eAAe,eAAGB,CAA/B,K;MACIB,IAAI,eAAc,WAAAd,QAA4B,CAAhC,C;QACI,cAAc,W;MACIB,IAAI,eAAc,UAAAd,QAA6B,CAAjC,C;QACI,cAAkB,cAAc,UAAIB,GAAgC,UAAhC,GAAmD,U;MACrE,OAAO,W;K;;IAZf,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;qDAgBA,qB;MAEI,WAvD,mBAAY,cAuenC,SAvemC,IAAZ,C;MAwehD,WAAe,kBAAa,cAAO,IAAxB,GAA8B,WAA9B,GAAwC,cAAO,kBAAY,OAAnB,I;MACnD,UAAU,IAAV,EAAGB,cAAhB,C;K;;IA5iBJ,iD;MAAA,oD;MAGwC,+B;MApB5C,sB;MAqBsB,Q;MACV,wBAAmB,CAAnB,C;QAAwB,4D;WACxB,sBAakB,CAAIB,C;QAAuB,uBAAa,eAAb,O;;QACf,MAAM,gCAAyB,uBAAoB,eAA7C,C;MAHIB,0B;MAJJ,Y;K;IAWA,kC;MAAA,oD;MAGoB,+B;MA/BxB,sB;MAGCQ,sBAAc,qD;MAJIB,Y;K;IAOA,4C;MAAA,oD;MAG2C,+B;MAtC/C,sB;MAuCCQ,sBtJpB8D,YsJoBhD,QtJpBgD,C;MsJqB9D,aAAO,mBAAY,O;MACnB,IAAI,mB5KsrPD,YAAQ,C4KtrPX,C;QAA2B,sBAAc,qD;MAN7C,Y;K;IC5BJ,4B;MAMoB,Q;M7Ky4rBA,U;MADhB,UAAe,C;MACf,uD;QAAgB,cAAhB,iB;QACI,YAAgB,O6K34rBiB,O7K24rBjC,I;;M6K34rBJ,aAAa,iB7K64rBN,G6K74rBM,C;MACb,wBAAGB,SAAhB,gB;QAAgB,gBAAA,SAAhB,M;QACW,SAAP,MAAO,EAAO,SAAP,C;;MAEX,OAAO,M;K;IAGX,0B;MASiB,Q;MAFb,YAAY,iBAAa,gBAAb,C;MACZ,YAAY,iBAAa,gBAAb,C;MACZ,wBAAA,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAAS,KAAT,C;K;gGAGX,qB;MAWW,4B;MAAA,U;QAAqB,OAAL,S7KirPhB,YAAQ,C;;M6KjrPf,W;K;oFAGJ,mC;MAUI,O7KoqPO,qBAAQ,C6KppPf,GAAe,cAAf,GAAmC,S;K;IAGvC,iD;MAMI,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAL,KAAa,KAAM,OAAXD,C;QAA8D,OAAO,K;MAErE,4C;QACI,SAAS,UAAK,CAAL,C;QACT,SAAS,MAAM,CAAN,C;QAET,IAAI,OAAO,EAAX,C;UACI,qeACG,IAAI,cAAc,UAAIB,C;UACH,OAAO,K;;QAIP,0BAAsB,kBAAtB,C;UAA4C,IAAI,CAAI,kBAAH,EAAG,EAakB,EAAlB,CAAR,C;YAA+B,OAAO,K;eACIF,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,E

AAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,C  
AAR,C;YAA2B,OAAO,K;eAC9E,6BAAsB,qBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;  
;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2  
B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAA  
O,K;eAC9E,gCAAsB,wBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;e  
AC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,i  
CAAsB,yBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAE9E,qCAAsB  
,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,sCAAsB,8BA  
AtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,oCAAsB,4BAAtB,  
C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,qCAAsB,6BAAtB,C;UA  
A4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAEtE,IAAI,YAAM,EAAN,CAAJ,C;  
UAAc,OAAO,K;;MAIrc,OAAO,I;K;IAGX,4C;MAKI,IAAI,iBAAJ,C;QAakB,OAAO,M;MACzB,aAAa,CAAK,e  
AAL,gBAAK,EAAa,SAAb,CAAL,GAA6C,CAA7C,QAAiD,CAAjD,I;MvC6SkB,kBAAXB,mBuC5SY,MvC4SZ,C  
;MuC3SH,oDxK5BgD,gBwK4BhD,C;MADJ,O3JnCO,WoH+U6C,W;K;IuCvSxD,mE;MAEI,IAAY,SAAR,0BAAJ  
,C;QACI,MAAO,gBAAO,OAAP,C;QACP,M;;MAEJ,SAAU,WAAI,SAAJ,C;MACV,MAAO,gBAAO,EAAP,C;M  
AEP,4C;QACI,IAAI,MAAK,CAAT,C;UACI,MAAO,gBAAO,IAAP,C;;QAEX,cAAc,UAAK,CAAL,C;QAEV,IAD  
E,OACF,S;UAAmB,MAAO,gBAAO,MAAP,C;aAC1B,mBAFE,OAEF,E;UAA2B,4BAAR,OAAQ,EAA4B,MAA5  
B,EAAoC,SAAP,C;aAC3B,uBAHE,OAGF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,wBAJE,  
OAlF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,sBALE,OAKF,E;UAAmB,MAAO,gBA Ae,gB  
AAR,OAAQ,CAAF,C;aAC1B,uBANE,OAMF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,wBAP  
E,OAOE,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,yBARE,OAQF,E;UAAmB,MAAO,gBA Ae,  
gBAAR,OAAQ,CAAF,C;aAC1B,uBATE,OASF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,0BA  
VE,OAUf,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAE1B,kBAZE,OAYF,c;UAAmB,MAAO,gBA Ae  
,kBAAR,OAAQ,CAAF,C;aAC1B,kBAeE,OAAf,e;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBA  
dE,OAcF,a;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBAfE,OAEf,c;UAAmB,MAAO,gBA Ae,k  
BAAR,OAAQ,CAAF,C;;UAEP,MAAO,gBAAO,OAAQ,WAAf,C;;MAIIC,MAAO,gBAAO,EAAP,C;MACP,SAA  
U,kBAAmB,iBAAV,SAAU,CAAnB,C;K;IxKjJd,yB;MAAA,6B;K;sCACI,Y;MAAkC,Y;K;OCACIC,Y;MAAsC,Y;  
K;wCACtC,Y;MAAGC,Q;K;4CACHC,Y;MAAoC,S;K;mCACpC,Y;MAA+B,MAAM,6B;K;uCACrC,Y;MAAmC,  
MAAM,6B;K;;IAN7C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;IASA,qB;MAAA,yB;MACI,+C;K;iCAEA,iB;M  
AA4C,qCAAoB,KAAm,U;K;mCACtE,Y;MAA+B,Q;K;mCAC/B,Y;MAAkC,W;K;iFAEX,Y;MAAQ,Q;K;kCAC/  
B,Y;MAAkC,W;K;yCACIC,mB;MAAmD,Y;K;8CACnD,oB;MAAmE,OAAA,QAAS,U;K;sCAE5E,iB;MAAwC,  
MAAM,8BAA0B,iDAA8C,KAA9C,MAA1B,C;K;wCAC9C,mB;MAA8C,S;K;4CAC9C,mB;MAAkD,S;K;mCAEI  
D,Y;MAA6C,kC;K;uCAC7C,Y;MAAqD,kC;K;+CACrD,iB;MACI,IAAI,UAAS,CAAb,C;QAAgB,MAAM,8BAA0  
B,YAAS,KAAnc,C;MACTB,OAAO,2B;K;OCAGX,8B;MACI,IAAI,cAAa,CAAb,IAAkB,YAAW,CAAjC,C;QAAo  
C,OAAO,I;MAC3C,MAAM,8BAA0B,gBAAa,SAAb,mBAAkC,OAA5D,C;K;wCAGV,Y;MAAiC,8B;K;;IA5BrC,  
iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IA+BA,iC;MAA8D,6BAAkB,SAAlB,EAAoC,KAApC,C;K;IAE5B,8C;MA  
AC,oB;MAA0B,0B;K;yFACIC,Y;MAAQ,OAAA,WAAO,O;K;OCACtC,Y;MAAkC,OAAA,WL4qP3B,YAAQ,C;K  
;iDK3qPf,mB;MAA6C,OAAO,SAAP,WAAO,EAAS,OAAT,C;K;sDACpD,oB;MAAsE,c;;QgBkoDtD,Q;QADhB,I  
AAI,ChBjoDyD,QgBioDzD,iBhBjoDyD,QgBioDnC,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,OhBloD6C,Qg  
BkoD7C,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,ChBloDkD,oBgBkoDvC,OhBloDuC,CgBkoDtD  
,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MhBnoDsD,iB;K;2CAC7D,Y;MAAuC,OAAO,qBAAP,WAAO,  
C;K;OCAC9C,Y;MAC+C,gBAAP,W;MAAA,OAAwB,cAAxB,GiBiKpC,SjBjKoC,GiBmKpC,SN63BoB,Q;K;;IX7  
hC5B,qB;MAIsC,8B;K;IAEtC,4B;MAIqD,OAAl,QAAS,OAAT,GAAgB,CAApB,GAAgC,OAAT,QAAS,CAAhC,  
GAA8C,W;K;mFAEnG,yB;MAAA,qD;MAAA,mB;QAK0C,kB;O;KAL1C,C;-FAOA,yB;MAAA,+D;MAAA,mB;  
QAMwD,uB;O;KANxD,C;2FAQA,yB;MAAA,+D;MAAA,mB;QAMoD,uB;O;KANpD,C;IAQA,mC;MAKI,OAAI  
,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAXB,GAAyC,iBAAU,sBAAkB,QAAIB,EAAwC,IAAxC,CAAV,C;K  
;IAE7C,iC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAXB,GAAyC,iBAAU,sBAAkB,QAAIB,EA  
AwC,IAAxC,CAAV,C;K;IAE7C,gC;MAI2D,OAAI,eAAJ,GAAqB,OAAO,OAAP,CAArB,GAA0C,W;K;IAErG,m

C;MAImE,OAAS,cAAT,QAAS,C;K;gFAE5E,yB;MAaA,gE;MAbA,6B;QAYBI,WAAW,eAduE,IAcvE,C;QaCX,iB  
AAc,CAAd,UbfkF,Iaef,U;UbA6B,eAf2D,IAevD,CaCtB,KbDsB,CAAJ,C;;QAFyC,OAGB/D,I;O;KA3BX,C;8FAA  
A,yB;MAAA,gE;MAAA,6B;QAYI,WAAW,eAAa,IAAb,C;QaCX,iBAAc,CAAd,UbAO,IaAP,U;UbA6B,eAAI,Ka  
CtB,KbDsB,CAAJ,C;;QAC7B,OAAO,I;O;KAdX,C;wFAiBA,yB;MiBzFA,+D;MjByFA,gC;QiBrF0B,gBAAf,gB;Q  
jBsGkB,aa5FzB,W;Qb4FA,Oa3FO,SIXoC,Q;O;KjBqF/C,C;yFAyBA,yB;MiB3GA,4E;MAAA,gE;MjB2GA,0C;Qi  
BvGI,qBjB4HyB,QiB5HzB,C;QAC8B,gBAAvB,ejB2HkB,QiB3HIB,C;QjB2H4B,aazHnC,W;QbyHA,OaxHO,SIH  
4C,Q;O;KjBsGvD,C;IAkCI,mC;MAAQ,uBAAG,iBAAO,CAAP,IAAH,C;K;IAQR,qC;MAAQ,OAAA,SAAK,KAA  
L,GAAY,CAAZ,I;K;4FAEZ,qB;MAK4D,QAAC,mB;K;kGAE7D,qB;MAWI,OAAO,qBAAGB,SAAK,U;K;sFAGh  
C,yB;MAAA,qD;MAAA,4B;QAKgE,uCAAQ,W;O;KALxE,C;sFAOA,yB;MAAA,qD;MAAA,4B;QAKoD,uCAA  
Q,W;O;KAL5D,C;sFAOA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,GAAmC,S;K;4FAGvC,+B;MAQoH,OAAA,SA  
AK,qBAAY,QAAC,C;K;IAGzH,uC;MAK+E,kBAAhB,0B;MAAwB,+B;MAAxB,Oa9MpD,W;K;lbiNX,yC;MAAk  
D,QAAM,cAAN,C;aAC9C,C;UAD8C,OACzC,W;aACL,C;UAF8C,OAEzC,OAAO,sBAAK,CAAL,CAAP,C;;UAF  
yC,OAGtC,S;;K;IAGZ,8D;MAgBkE,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACjG,WAAW,cAAX,E  
AAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,  
C;QACI,UAAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,cAAc,MAAd,EA  
AsB,OAAtB,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UA  
CD,OAAO,MAAM,CAAN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,4E;MAe8E  
,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC7G,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;M  
AEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAY,GAAN,IA  
AM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,UAAW,SAAQ,MAAR,EAAgB,OAAhB,C;QAErB,I  
AAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CA  
AN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;kGAGX,yB;MAAA,8D;MAAA,4D;MAS  
BqC,8D;QAAA,qB;UAAE,qBAAc,iBAAS,EAAT,CAAd,EAA4B,WAA5B,C;S;O;MatBvC,+D;QAKBI,yB;UAAA  
,YAAiB,C;QACjB,uB;UAAA,UAAe,c;QAGf,+BAaA,SAAb,EAAwB,OAAxB,EAAiC,oCAAjC,C;O;KAtBJ,C;IA6  
BA,mE;MAmBoC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACnE,WAAW,cAAX,EAAiB,SAAjB,EAA  
4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,G  
AAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,WAAW,MAAX,C;QAEV,IAAI,MA  
AM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;;U  
AEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,8C;MAMQ,gBAAY,OAAZ,C;QAAuB,M  
AAM,gCAAYB,gBAaA,SAAb,mCAAkD,OAAID,OAAzB,C;WAC7B,gBAAY,CAAZ,C;QAAiB,MAAM,8BAA0B  
,gBAaA,SAAb,yBAA1B,C;WACvB,cAAU,IAAV,C;QAAkB,MAAM,8BAA0B,cAAW,OAAZ,gCAA2C,IAA3C,  
OAA1B,C;K;IAchC,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;IAE1C,8B;MAEoC,MAAM,wBAAoB,8BAAp  
B,C;K;;;wF0Gnb1C,yB;MzGgCA,wE;MyGhCA,uC;QAmBW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QA  
Af,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UACIB,Wz  
GyKJ,ayGzKgB,GzGyKhB,EyGvMyC,SA8BIB,CAAU,GAAY,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uBAAuB,  
CAAC,WAAy,mBAAY,GAAZ,CAAnE,CzGyKvB,C;;QyGvMA,OAGCO,W;O;KAnDX,C;4FAsBA,6C;MAwBc,  
Q;MAAA,OAAA,SAAK,iB;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAaK  
B,sBAAY,GAAZ,C;QACIB,WzGyKJ,ayGzKgB,GzGyKhB,EyGzKuB,UAAU,GAAY,EAAe,WAAf,EAA4B,CAA  
5B,EAA+B,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAAnE,CzGyKvB,C;;MyGvKA,OAAO,W;K;iFAGX,yB;M  
AAA,gB;MAAA,8B;MzGtBA,wE;MyGsBA,6D;QAnCW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QAAf,O  
AAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UA8BwE,U;UA7  
B1F,WzGyKJ,ayGzKgB,GzGyKhB,EyG5IkC,UA7BD,GA6BC,EA7BoB,uBAAuB,CAAC,WAAy,mBAAY,GAA  
Z,CA6BzC,GAAY,qBA7B3B,GA6B2B,EA7BT,CA6BS,CAAX,GAA6C,UA7BxD,WA6BwD,6DAA5D,EA7BiB,  
CA6BjB,CzG4IIC,C;;QyG7IA,OA1BO,W;O;KAGX,C;kFA0BA,yB;MAAA,gB;MAAA,8B;MAAA,0E;QAIcC,Q;  
QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6DQ,WA7D  
U,WAAy,GAAZ,C;UA6DuF,U;UAAjG,WzG6GZ,ayGzKgB,GzGyKhB,EyG7GiD,UA5DhB,GA4DgB,EA5DK,u  
BAAuB,CA4DjE,WA5D8E,mBAAY,GAAZ,CA4D1B,GAAY,qBA5D1C,GA4D0C,EA5DxB,CA4DwB,CAAX,G  
AA6C,UA5DvE,WA4DuE,6DAA5D,EA5DE,CA4DF,CzG6GjD,C;;QyG9GA,OACY,W;O;KA7BhB,C;iFAgCA,y

B;MAAA,gB;MAAA,8B;MzGhFA,wE;MyGgFA,qD;QA7FW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QA Af,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKb,sBAAY,GAAZ,C;UakFiD,U; UAjFnE,WzGyKJ,ayGzKgB,GzGyKhB,EyGxFgC,UAjFsB,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAiFhD,kB AA6B,UAjFjC,WaIFiC,6DAAvC,EAjFmB,CAiFnB,CzGwFhC,C;;QyGzFA,OA9EO,W;O;KA6DX,C;oFAoBA,yB ;MAAA,gB;MAAA,8B;MAAA,kE;QAtFc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU ,sBAAM,CAAN,C;UACV,kBA2GQ,WA3GU,WAAy,GAAZ,C;UA2GgE,U;UAA1E,WzG+DZ,ayGzKgB,GzGyK hB,EyG/D+C,UA1GO,uBAAuB,CA0GjE,WA1G8E,mBAAY,GAAZ,CA0GjC,kBAA6B,UA1GhD,WA0GgD,6DA AvC,EA1GI,CA0GJ,CzG+D/C,C;;QyGhEA,OACY,W;O;KAvBhB,C;qFA0BA,yB;MAAA,gB;MAAA,8B;MzG9H A,wE;MyG8HA,uC;QA3IW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;U ACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKb,sBAAY,GAAZ,C;UACC,oB;UakIc,U;UAAjC,IAIikD,uBAAuB, CAAC,WAAy,mBAAY,GAAZ,CAkItF,C;YADA,mBAjI+C,C;;YAiI/C,mBACKB,UAIIW,GakIX,EAAe,UAIIC, WAKID,6DAAf,EAI16B,CAkI7B,C;;UAI1IB,WzGyKJ,ayGzKgB,GzGyKhB,mB;;QyGzCA,OA9HO,W;O;KA2GX, C;sFAwBA,yB;MAAA,gB;MAAA,8B;MAAA,oD;QAxIc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB ;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6JQ,WA7JU,WAAy,GAAZ,C;UACC,oB;UA8Jc,U;UAAjC,IA9Jk D,uBAAuB,CA4JjE,WA5J8E,mBAAY,GAAZ,CA8JtF,C;YADA,mBA7J+C,C;;YA6J/C,mBACKB,UA9JW,GA8JX ,EAAe,UA9JC,WA8JD,6DAAf,EA9J6B,CA8J7B,C;;UAFV,WzGaZ,ayGzKgB,GzGyKhB,mB;;QyGbA,OAAY,W; O;KAvBhB,C;IA6BA,6C;MArKc,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAA M,CAAN,C;QACV,kBA+KG,WA/Ke,WAAy,GAAZ,C;QA2GgE,U;QAOE/E,WzGLP,ayGzKgB,GzGyKhB,EyGK mC,CA9KmB,uBAAuB,CA8KtE,WA9KmF,mBAAY,GAAZ,CA0GjC,GAoErC,CAPeQc,GAA6B,UA1GhD,WA0 GgD,6DAoEnD,IAAM,CAAN,IzGLnC,C;;MyGKA,OAAO,W;K;I+DnPOB,oC;MAAC,kB;MAAuB,kB;K;;wCAN7 D,Y;MAMsC,iB;K;wCANtC,Y;MAM6D,iB;K;0CAN7D,wB;MAAA,wBAMsC,qCANtC,EAM6D,qCAN7D,C;K;s CAAA,Y;MAAA,OAMsC,mDANtC,IAM6D,wCAN7D,O;K;sCAA,Y;MAAA,c;MAMsC,sD;MAAuB,sD;MAN7 D,a;K;oCAA,iB;MAAA,4IAMsC,sCANtC,IAM6D,sCAN7D,I;K;wFjKEA,yB;MAAA,kC;MAAA,4C;MAAA,kD ;QAMuF,wC;O;MANvF,4CAOI,Y;QAAuC,8B;O;MAP3C,8E;MAAA,2B;QAMuF,2C;O;KANvF,C;IACsC,2C;MA AC,wC;K;0CACnC,Y;MAAqD,4BAAiB,wBAAjB,C;K;;IAIzD,yC;MAI4D,OAAI,oCAAJ,GAA2B,SAAK,KAAhC ,GAA0C,I;K;IAEtG,uD;MAI0E,OAAI,oCAAJ,GAA2B,SAAK,KAAhC,GAA0C,S;K;IAGpH,8B;MAMoB,Q;MAD hB,aAAa,gB;MACG,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACL,OAAP,MAAO,EAAO,OAAP,C;;MAEX, OAAO,M;K;IAGX,4B;MAUiB,Q;MAHb,mBAAmB,mCAAwB,EAAxB,C;MACnB,YAAy,iBAAa,YAAb,C;MAC Z,YAAy,iBAAa,YAAb,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;Q ACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAAS,KAAT,C;K;wFUxDX,qB;MAKqE,gB;K;IAErE,iC;M AMoE,4BAAiB,SAAjB,C;K;uFAEpE,gC;MAKI,OAAGB,mBAAhB,C;QAAgB,8B;QAAM,UAAU,OAAY,C;;K;IA MY,oC;MAAC,0B;MACnC,eAAoB,C;K;yCACpB,Y;MAAwC,OAAA,eAAS,U;K;sCACjD,Y;MAA6E,Q;MAAhC ,wBAAa,oBAAmB,mBAAnB,EAAmB,2BAAnB,QAAb,EAA0C,eAAS,OAAnD,C;K;;sFwJ5BjD,yB;MAAA,4E;M AAA,gB;MAAA,8B;MAAA,+C;QAUiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAVjC,C;wFAYA,y B;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAWiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KA XjC,C;sFAaA,+C;MAQI,SAAK,aAAI,QAAS,aAAb,EAAmB,KAAAnB,C;K;ICnCT,8C;MAUI,IAAI,wCAAJ,C;QA CI,OAAO,SAAK,4BAAqB,GAARb,C;MAET,4B;M1KuTI,Q;MALX,YAAy,oB0KITa,G1KkTb,C;MACZ,IAAI,iB AAiB,CAAC,4B0KnTG,G1KmTH,CAAtB,C;Q0KnTgC,MAAM,2BAAuB,wCAAvB,C;;Q1KuTIC,2BAAO,sE;;M 0KvTX,+B;K;IAGJ,8C;MAUQ,kBADE,SACf,kB;QADJ,OACkC,YAAT,SAAK,IAAI,EAAY,YAAZ,C;;QADIC,O AEY,uBAAmB,SAAnB,EAAyB,YAAzB,C;K;IAGhB,gD;MAWQ,kBADE,SACf,yB;QADJ,OACyC,cAAT,SAAK ,IAAI,EAAY,YAAZ,C;;QADzC,OAey,8BAA0B,SAAI1B,EAAgC,YAAhC,C;K;;;;;IAc0B,4C;MAAC,wB;MAAo C,0B;K;qEAApC,Y;MAAA,yB;K;0CACvC,iB;MAA4C,OAAI,OA AJ,QA AI,EAAO,KAAP,C;K;4CACHD,Y;MAA +B,OAAI,SAAJ,QAAL,C;K;4CACnC,Y;MAAkC,OAAA,QAAL,W;K;0FACf,Y;MAAQ,OAAA,QAAL,K;K;2CACn C,Y;MAAkC,OAAA,QAAL,U;K;qDACtC,e;MAA4C,OAAA,QAAL,mBAAY,GAAZ,C;K;uDACHD,iB;MAAgE,O AAA,QAAL,qBAAC,KAAAd,C;K;6CACpE,e;MAA+B,OAAA,QAAL,WAAI,GA AJ,C;K;0FACT,Y;MAAQ,OAAA, QAAL,K;K;4FACH,Y;MAAQ,OAAA,QAAL,O;K;6FACJ,Y;MAAQ,OAAA,QAAL,Q;K;8DAEvD,e;MAAmD,gBA AJ,Q;MAAI,4B;M1K+PxC,Q;MALX,YAAy,oB0K1PyD,G1K0PzD,C;MACZ,IAAI,iBAAiB,CAAC,4B0K3P+C,G 1K2P/C,CAATB,C;QACI,2B0K5PwE,mB;;Q1K+PxE,2BAAO,sE;;M0K/PoC,+B;K;;IAGN,mD;MAAC,wB;MAA2

C,0B;K;4EAA3C,Y;MAAA,yB;K;iDAC1C,iB;MAA4C,OAAI,OAAJ,QAAI,EAAO,KAAP,C;K;mDACHd,Y;MAA+B,OAAI,SAAJ,QAAI,C;K;mDACnC,Y;MAAkC,OAAA,QAAI,W;K;iGACf,Y;MAAQ,OAAA,QAAI,K;K;kDACHnC,Y;MAAkC,OAAA,QAAI,U;K;4DACtC,e;MAA4C,OAAA,QAAI,mBAA,Y,GAZ,C;K;8DACHd,iB;MAAgE,OAAA,QAAI,qBAAC,KAAd,C;K;oDACpE,e;MAA+B,OAAA,QAAI,WAAI,GAJ,C;K;iGACF,Y;MAAQ,OAAA,QAAI,K;K;mGACH,Y;MAAQ,OAAA,QAAI,O;K;oGACU,Y;MAAQ,OAAA,QAAI,Q;K;sDAE5E,sB;MAAyC,OAAA,QAAI,aAAI,GAJ,EAAS,KAAT,C;K;uDAC7C,e;MAAkC,OAAA,QAAI,cAAO,GAAP,C;K;yDACtC,gB;MAA2C,QAAI,gBAAO,IAAP,C;K;gDAC/C,Y;MAAuB,QAAI,Q;K;qEAE3B,e;MAAmD,gBAAJ,Q;MAAI,4B;M1K00xC,Q;MALX,YAAY,oB0KrOyD,G1KqOzD,C;MACZ,IAAI,iBAAiB,CAAC,4B0KtO+C,G1KsO/C,CAAtB,C;QACI,2B0KvOwE,mB;;Q1K00xE,2BAAO,sE;;M0K10oC,+B;K;;I1KvFnD,oB;MAAA,wB;MACI,8C;K;gCAEA,iB;MAA4C,oCAAsB,KAAM,U;K;kCACxE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K;iCAC/B,Y;MAAkC,W;K;2CAEIC,e;MAA+C,Y;K;6CAC/C,iB;MAAsD,Y;K;mCACtD,e;MAAwC,W;K;mFACY,Y;MAAQ,6B;K;gFAC/B,Y;MAAQ,6B;K;kFACI,Y;MAAQ,8B;K;uCAEjD,Y;MAAiC,6B;K;;IAjBrC,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IAoBA,oB;MAMuE,Q;MAA7B,OAA6B,uE;K;IAEvE,wB;MAaI,OAAI,KAAM,OAAN,GAAa,CAAjB,GAAOB,QAAN,KAAM,EAAM,qBAAC,YAAY,KAAM,OAAIB,CAAd,CAAN,CAA1B,GAA6E,U;K;kFAEjF,yB;MAAA,oD;MAAA,mB;QAO8C,iB;O;KAP9C,C;8FASA,yB;MAAA,wE;MAAA,mB;QAQ4D,2B;O;KAR5D,C;IAUA,+B;MAyID,gBAA7C,qBAAoB,YAAY,KAAM,OAAIB,CAApB,C;MAAQD,wB;MAArD,OYJO,S;K;wFZMX,yB;MAAA,4D;MAAA,mB;QAOSD,qB;O;KAPtD,C;IASA,4B;MAM8G,gBAAvC,eAAc,YAAY,KAAM,OAAIB,CAAd,C;MAA+C,wB;MAA/C,OYrB5D,S;K;4FZuBX,yB;MAAA,wE;MAAA,mB;QAK8D,2B;O;KAL9D,C;IAOA,8B;MAU+E,OAAM,QAAN,KAAM,EAAM,qBAAC,YAAY,KAAM,OAAIB,CAAd,CAAN,C;K;sFAErF,yB;MgBfA,wE;MhBeA,gC;QgBXiC,gBAAtB,oB;QhB8BiB,aY9DxB,W;QZ8DA,OY7DO,SI+B2C,Q;O;KhBwTD,C;uFA2BA,yB;MgBnCA,uE;MhBmCA,0C;QgB/ByC,gBAA9B,mBhBsDiB,QgBtDjB,C;QhBsD2B,aY7FIC,W;QZ6FA,OY5FO,SlScmD,Q;O;KhB+B9D,C;4FAqCA,qB;MAK+D,QAAC,mB;K;kGAHE,qB;MAWI,OAAO,qBAAGB,mB;K;sFAG3B,yB;MAAA,oD;MAAA,4B;QAM2D,uCAAQ,U;O;KANnE,C;sFAQA,mC;MASI,OAAI,mBAAJ,GA Ae,cAAf,GAAmC,S;K;yFAEvC,yB;MAyBA,kC;MAAA,8B;MAZBA,iC;QAgCiC,Q;QAx2E,OAwbxD,CAAnB,wDAAmB,oBAxB0E,GAwBpE,C;O;KAhCpD,C;+EAUA,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAKiC,Q;QAA7B,OAAgD,CAAnB,wDAAmB,YAAI,GAJ,C;O;KALpD,C;+EAOA,iC;MAKI,sBAAI,GAJ,EAAS,KAAT,C;K;4FAGJ,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAOiC,Q;QAA7B,OAAgD,CAAnB,wDAAmB,oBAA,Y,GAZ,C;O;KAPpD,C;gGASA,4B;MASsG,OAAA,SAAK,qBAAC,KAAd,C;K;kFAG3G,yB;MAAA,gD;MAAA,8B;MAAA,iC;QASiC,Q;QAA7B,OAAuD,CAA1B,+DAA0B,eAAO,GAAP,C;O;KAT3D,C;6FAWA,qB;MAWoE,oB;K;6FAEpE,qB;MAWoE,sB;K;kFAEpE,yB;MAAA,6B;MAAA,4B;QAIgE,qBAAK,aAAL,EAAU,eAAV,C;O;KAJhE,C;2FAMA,wC;MAOiF,Q;MAAA,mCAAI,GAJ,oBAA,Y,c;K;uGAG7F,yB;MAAA,gB;MAAA,8B;MAAA,+C;QAMe,Q;QALX,YAAY,oBAAI,GAJ,C;QACZ,IAAI,iBAAiB,CAAC,4BAA,Y,GAZ,CAAtB,C;UACI,OAAO,c;;UAGP,OAAO,sE;;O;KANf,C;IAUA,oC;MAUkD,uCAAqB,GAArB,C;K;sFAEID,wC;MAWW,Q;MADP,YAAY,oBAAI,GAJ,C;MACL,IAAI,aAAJ,C;QACH,aAAa,c;QACb,sBAAI,GAJ,EAAS,MAAT,C;QACA,a;QAEA,Y;;MALJ,W;K;wFASJ,qB;MAMwF,OAAA,iBAAQ,W;K;wFAEHg,qB;MAMgH,OAAA,iBAAQ,W;K;4FAEXH,6C;Mem1BoB,Q;MAAA,Of90BT,iBe80BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qf90Ba,We+0Bb,aAAGB,Of/0Be,Ie+0B/B,Ef/0BsC,Se+0BZ,CAAe,OAAf,CAA1B,C;;Mf/0BhB,OAA6B,W;K;wFAGjC,6C;Me20BoB,Q;MAAA,Ofn0BT,iBem0BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qfn0Ba,Weo0Bb,afp0B0B,Seo0BtB,CAAY,OAAZ,CAAJ,EAAyC,Ofp0BC,Meo0B1C,C;;Mfp0BhB,OAA6B,W;K;IAGjC,kC;MAIyB,Q;MAArB,wBAAqB,KAArB,gB;QAAqB,aAAA,KAArB,M;QAAK,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAJ,EAAS,KAAT,C;;K;wFAIR,yB;MAAA,0D;MAAA,uE;MAAA,uC;QASW,kBAA,Y,mBAAoB,YAAY,cAAZ,CAApB,C;Qe4xBH,Q;QAAA,Of90BT,iBe80BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Uf90Ba,We+0Bb,aAAGB,Of/0Be,Ie+0B/B,Ef7xB2C,Se6xBjB,CAAe,OAAf,CAA1B,C;;Qf7xBhB,OAID6B,W;O;KAyCjC,C;oFAYA,yB;MAAA,0D;MAAA,uE;MAAA,uC;QAYW,kBAAU,mBAAoB,YAAY,cAAZ,CAApB,C;Qe6wBD,Q;QAAA,Ofn0BT,iBem0BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ufn0Ba,Weo0Bb,af9wByC,Se8wBrC,CAAY,OAAZ,CAAJ,EAAyC,Ofp0BC,Meo0B1C,C;;Qf9wBhB,OAtd6B,W;O;KA0CjC,C;0FAeA,yB;MAAA,w

E;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3FsE,QAAQ,W;QA2F5F,OAAC,cAAAd,C;UAAc,uB;U  
ACV,IAAI,UAAU,KAAM,IAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,  
OAAO,M;O;KAbX,C;8FAGBA,yB;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3GsE,Q  
AAQ,W;QA2G5F,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,KAAM,MAAhB,CAAJ,C;YACI,MAAO,aAAI,  
KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;KAbX,C;yFAiBA,6C;MAOoB,Q;MAAA,OAAA,SA  
3HoE,QAAQ,W;MA2H5F,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,WAAy,aA  
AI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,W;K;qFAGX,yB;MAAA,wE;MAAA,uC;QAOW,k  
BAAS,oB;QAFa,Q;QAAA,OA3HoE,iBAAQ,W;QA2H5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAcmC,SAd/B,C  
AAU,OAAV,CAAJ,C;YACI,WAAy,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX  
,C;+FAUA,6C;MAOoB,Q;MAAA,OAAA,SAPJoE,QAAQ,W;MAoJ5F,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAA  
I,CAAC,UAAU,OAAV,CAAL,C;UACI,WAAy,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,  
W;K;2FAGX,yB;MAAA,wE;MAAA,uC;QAOW,kBAAY,oB;QAFh,Q;QAAA,OA3HoE,iBAAQ,W;QAoJ5F,OAAg  
B,cAAhB,C;UAAgB,yB;UACZ,IAAI,CaCkC,SAdjC,CAAU,OAAV,CAAL,C;YACI,WAAy,aAAI,OAAQ,IAAZ,E  
AAiB,OAAQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX,C;IAUA,0B;MAQqB,IAAN,I;MADX,IAAI,oCAAJ,C;QA  
CW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAaKb,sBAaK,CAAL,CAAI  
B,GAA+B,oBAAW,OAAdD,C;YAAL,K;;YACQ,0BAAM,qBAAoB,YAAY,cAAZ,CAApB,CAAN,C;YAHL,K;;Q  
AAP,W;;MAMJ,OAAoC,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAGxC,yC;MAIwB,SAApB,WAAoB,Y;MAA  
pB,kB;K;IAEJ,4B;MAM6D,QAAM,gBAAN,C;aACzD,C;UADyD,OACpD,U;aACL,C;UAFyD,OAEPD,MAAM,U  
AAK,CAAL,CAAN,C;;UAFoD,OAGjD,mBAAM,qBAAoB,YAAY,gBAAZ,CAApB,CAAN,C;;K;IAGZ,yC;MAI  
wB,OAAPB,WAAoB,Y;MAAPB,kB;K;IAEJ,4B;MAM4D,OAA6B,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAEz  
F,yC;MAIwB,SAAPB,WAAoB,Y;MAAPB,kB;K;IAEJ,4B;MAMqD,QAAM,cAAN,C;aACjD,C;UADiD,OAC5C,U  
;aACL,C;UAFiD,OgBIY8B,uB;;UhBkY9B,OAGzC,uB;;K;IAGZ,iC;MAMmE,4BAAC,SAAd,C;K;IAEnE,yC;MAK  
I,WAAoB,0B;MAAPB,kB;K;IAEJ,kC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAAoB,MAAM,IAAN,C;;  
QAAqC,kBAAPB,qBAAC,SAAd,C;QAA4B,wBAAS,UAAT,EAAqB,WAArB,C;QAAjE,OYpiBO,W;;MZoiBP,W;  
K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,MAAN,KAAM,C;;QAAiC,kBAAPB,qBAAC  
,SAAd,C;QAA4B,4B;QAAnE,OY7iBO,W;;MZ6iBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAA  
A,OAA0B,QAAN,KAAM,C;;QAAiC,kBAAPB,qBAAC,SAAd,C;QAA4B,0B;QAAnE,OYtjBO,W;;MZsjBP,W;K;I  
AEJ,mC;MAOWB,kBAAPB,qBAAC,SAAd,C;MAA4B,4B;MAA5B,OAA4C,oBY/jBrC,WZ+jBqC,C;K;IAEhD,iC;  
MAOWB,kBAAPB,qBAAC,SAAd,C;MAA4B,+B;MAA5B,OYxkBO,W;K;0FZ2kBX,2B;MAKI,sBAAI,IAAK,MA  
AT,EAAgB,IAAK,OAARB,C;K;4FAGJ,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,y  
B;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,  
KAAP,C;O;KALJ,C;4FAQA,0B;MAKI,yBAAO,GAAP,C;K;IAGJ,kC;MAOWB,kBAAf,aAAL,SAAK,C;MAcCL,6  
B;MAAtCA,OAA+C,oBY1nBxC,WZ0nBwC,C;K;IAEnD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAqCK,YAAL,gB  
AAK,O;MArCV,OAAgD,oBYpoBzC,WZooByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAoCK,YAA  
L,gBAAK,O;MApCV,OAAgD,oBY9oBzC,WZ8oByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAMC  
K,YAAL,gBAAK,O;MANCV,OAAgD,oBYxpBzC,WZwpByC,C;K;4FAEpD,0B;MAMI,uBAAO,GAAP,C;K;8FA  
GJ,yB;MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KAAC,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;M  
AAA,kC;QAMc,UAAV,SAAK,KAAC,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,kC;QAMc,UA  
AV,SAAK,KAAC,EAAU,IAAV,C;O;KANd,C;IAUA,wC;MACsD,QAAM,cAAN,C;aACID,C;UADkD,OAC7C,U;  
aACL,C;UAFkD,gB;;UAAA,OAG1C,S;;K;oF2K5wBZ,yB;MAAA,8D;MAAA,8B;MAAA,qC;QAUIC,Q;QAA7B,  
OAA2D,CAA9B,sEAA8B,eAAO,OAAP,C;O;KAV/D,C;wFAYa,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASiC,Q;  
QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;wFAWA,yB;MAAA,8D;MAAA,8B;MAAA,s  
C;QASiC,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;4FAWA,8B;MAKI,SAAK,WAAI,  
OAAJ,C;K;4FAGT,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MA  
AA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAAA,sC;QAKS,O  
AAL,SAAK,EAAO,QAAP,C;O;KALT,C;8FAQA,8B;MAKI,SAAK,cAAO,OAAP,C;K;8FAGT,yB;MAAA,sD;MA  
AA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAA  
K,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KAL

T,C;IAQA,qC;MAIU,IAIe,I;MAHjB,kBADE,QACF,c;QAAiB,OAAO,yBAAO,QAAP,C;;QAEpB,aAAsB,K;QAC  
T,0B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,oBAAI,IAAJ,CAAJ,C;YAAe,SAAS,I;;QAC5B,OAAO,M;;K;I  
AKnB,uC;MAKiB,Q;MADb,aAAsB,K;MACT,0B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,oBAAI,IAAJ,CA  
AJ,C;UAAe,SAAS,I;;MAE5B,OAAO,M;K;IAGX,uC;MAII,OAAO,yBAAGB,OAAT,QAAS,CAAhB,C;K;IAGX,i  
D;MAKI,OAAI,oCAAJ,GAAwB,SAAxB,GAAC,mB;K;IAEtC,0C;MAII,OAAO,4BAAmB,6BAAT,QAAS,CAAn  
B,C;K;IAGX,0C;MAII,WAAoB,UAAAT,QAAS,C;MACpB,O5KuEwD,C4KvEjD,I5KuEkD,U4KvElD,IAAqB,4BA  
AU,IAAV,C;K;IAGhC,0C;MAII,OjLuoPO,EiLvoPA,QjL+jPA,YAAQ,CAwER,CiLvoPA,IAAyB,4BAAmB,OAA  
T,QAAS,CAAnB,C;K;IAGpC,0C;MAII,OAAO,4BAAmB,6BAAT,QAAS,CAAnB,C;K;IAGX,0C;MAII,IjLynPO,  
EiLznPH,QjLijPG,YAAQ,CAwER,CiLznPP,C;QACI,OAAO,4BAAmB,OAAT,QAAS,CAAnB,C;;QAEp,OAAO,  
wB;K;IAGf,0C;MAII,WAAoB,UAAAT,QAAS,C;MACpB,I5KuCwD,C4KvCpD,I5KuCqD,U4KvCzD,C;QACI,OA  
AO,4BAAU,IAAV,C;;QAEp,OAAO,wB;K;IAGf,kC;MACI,a5KgCwD,CAAC,mB;M4K/BzD,iB;MACA,OAAO,  
M;K;IAIX,2C;MAKkF,gCAAc,SAAd,EAAYB,IAAZB,C;K;IAEIF,2C;MAKkF,gCAAc,SAAd,EAAYB,KAAZB,C;K  
;IAEIF,sE;MACI,iBAAa,KAAb,C;M/JvJgB,kB+JwJX,oB;MACD,OAAO,qBAAP,C;QACI,IAAI,UAAU,kBAAV,6  
BAAJ,C;UACI,oB;UACA,WAAS,I;;MAGrB,OAAO,Q;K;oFAIX,4B;MAM6D,kCAAS,KAAAT,C;K;IAE7D,gC;M  
AKiD,IAAI,mBAAJ,C;QAAe,MAAM,2BAAUb,gBAAvB,C;;QAArB,OAAMe,2BAAS,CAAT,C;K;IAEPH,sC;MA  
KwD,OAAI,mBAAJ,GAAe,IAAf,GAAYB,2BAAS,CAAT,C;K;IAEjF,+B;MAKgd,IAAI,mBAAJ,C;QAAe,MAA  
M,2BAAUb,gBAAvB,C;;QAArB,OAAMe,2BAAS,2BAAT,C;K;IAEnH,qC;MAKuD,OAAI,mBAAJ,GAAe,IAAf,  
GAAYB,2BAAS,2BAAT,C;K;IAEHF,2C;MAK8E,kCAAc,SAAd,EAAYB,IAAZB,C;K;IAE9E,2C;MAK8E,kCAAc,  
SAAd,EAAYB,KAAZB,C;K;IAE9E,wE;MAEgB,UAGS,MAHT,EAcy,MAdz,EAc6B,M;MAfzC,IAAI,uCAAJ,C;  
QACI,OAAoC,cAA5B,sEAA4B,EAAC,SAAd,EAAYB,uBAAZB,C;MAExC,iBAAsB,C;MACD,oC;MAArB,qBAA  
kB,CAAIb,mC;QACI,cAAc,sBAAK,SAAL,C;QACd,IAAI,UAAU,OAAV,MAAsB,uBAA1B,C;UACI,Q;QAEJ,IA  
AI,eAAc,SAAIb,C;UACI,sBAAK,UAAAL,EAAMb,OAAnB,C;QAEJ,+B;;MAEJ,IAAI,aAAa,cAAjB,C;QACwB,oC  
;QAAiB,mB;QAArC,oE;UACI,2BAAS,WAAT,C;QAEJ,OAAO,I;;QAEp,OAAO,K;;K;ICtSf,wB;K;kCAEI,Y;MA  
A4B,sB;K;;IAMhC,wB;K;kCAEI,Y;MAA4B,mC;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,uB;K;iCAEI  
,Y;MAA4B,qB;K;;IAMhC,wB;K;kCAEI,Y;MAA4B,sB;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,0B;K;  
oCAEI,Y;MAA4B,wB;K;;IAMhC,2B;K;qCAEI,Y;MAA4B,yB;K;;ICzDc,wC;MAAKC,uB;MAAjC,0B;K;4FACpB,  
Y;MAAQ,OAAA,eAAS,K;K;iDACxC,iB;MAAKC,mCAAS,0BAAoB,KAApB,CAAT,C;K;;IAGT,gC;MAAyC,8B;  
MAAxC,0B;K;oFACH,Y;MAAQ,OAAA,eAAS,K;K;yCACxC,iB;MAAKC,mCAAS,0BAAoB,KAApB,CAAT,C;K  
;mCAEIC,Y;MAAUb,eAAS,Q;K;8CAChC,iB;MAAUc,OAAA,eAAS,kBAAS,0BAAoB,KAApB,CAAT,C;K;yCA  
EhD,0B;MAA8C,OAAA,eAAS,aAAI,0BAAoB,KAApB,CAAJ,EAAGC,OAAhC,C;K;yCACvD,0B;MACI,eAAS,a  
AAI,2BAAqB,KAArB,CAAJ,EAAiC,OAAjC,C;K;;IAIjB,+C;MACoB,Q;MAAA,kC;MAAhB,IAAa,CAAT,0BAAJ  
,C;QAAA,OAA2B,8BAAY,KAAZ,I;;QAAuB,MAAM,8BAA0B,mBAAGB,KAAhB,2BAA0C,gBAAG,2BAAH,C  
AA1C,OAA1B,C;K;IAE5D,gD;MACoB,Q;MAAA,qB;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAAsB,iBAAO,K  
AAP,I;;QAAkB,MAAM,8BAA0B,oBAAiB,KAAjB,2BAA2C,gBAAG,cAAH,CAA3C,OAA1B,C;K;IAGID,+B;M  
AK+C,gCAAqB,SAArB,C;K;IAE/C,iC;MAM6D,wBAAa,SAAb,C;K;;;IrkPc7D,oD;MAQuF,wC;K;IARvF,8CAS  
I,Y;MAAUc,8B;K;IAT3C,gF;IsKa8G,wC;MAAA,mB;QAAE,kBAAS,aAAT,C;O;K;IAVhH,yB;MAUqG,oCAAS,s  
BAAT,C;K;IAErG,2B;MASI,eAAe,6B;MACf,oBAA0B,+BAAN,KAAM,EAawC,QAAxC,EAA+D,QAA/D,C;MA  
C1B,OAAO,Q;K;IAc+B,yB;K;+CAoBtC,kC;MAOI,IAAI,uCAA0B,QAAS,UAAvC,C;QAAkD,M;MACID,OAAO,  
sBAAS,QAAS,WAAIB,e;K;+CAGX,kC;MAQqD,6BAAS,QAAS,WAAIB,e;K;;;IAYzD,mC;MAA2C,wB;MAC  
vC,eAAoB,C;MACpB,mBAA4B,I;MAC5B,sBAAyC,I;MACzC,gBAAoC,I;K;gDAEpC,Y;MACI,OAAO,IAAP,C;  
QACI,QAAM,YAAN,C;eACI,C;YAAA,K;eACA,C;YACI,IAAI,kCAAe,UAAAnB,C;cACI,eAAQ,C;cACR,OAAO,I  
;;cAEP,sBAAe,I;;YALvB,K;eAOA,C;YAAc,OAAO,K;eACrB,C;eAAA,C;YAAgC,OAAO,I;;YAC/B,MAAM,yB;;  
QAGIB,eAAQ,C;QACR,WAAW,4B;QACX,gBAAW,I;QACX,I5HpFR,oBDgDQ,W6HoCY,kB7HpCZ,CChDR,C;  
;K;6C4HwFA,Y;MACU,IASe,I;MATrB,QAAM,YAAN,C;aACI,C;aAAA,C;UAAcC,OAAO,qB;aAC7C,C;UACI,e  
AAQ,C;UACR,OAAO,kCAAe,O;aAE1B,C;UACI,eAAQ,C;UACR,aACa,mF;UACb,mBAAY,I;UACZ,OAAO,M;;  
UAEH,MAAM,yB;;K;uDAItB,Y;MACI,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;;QAA8B,OAAO,W;K;2DAG/  
D,Y;MAA4C,QAAM,YAAN,C;aACxC,C;UADwC,OAC1B,6B;aACd,C;UAFwC,OAExB,6BAAsB,sBAAtB,C;;U  
AFwB,OAGhC,6BAAsB,uCAAoC,YAA1D,C;;K;IAOqC,4E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;

K;oDALR,+B;MACI,mBAAY,K;MACZ,eAAQ,C;MACR,OAA6C,0CAAtC,c;K;IAUsC,+E;MAAA,oB;QACzC,w  
CAAW,C;QAAX,OACA,yB;O;K;yDANR,kC;MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,M;MACzB,sBA Ae,Q;  
MACf,eAAQ,C;MACR,OAA6C,6CAAtC,c;K;2DAMX,kB;M7HNO,Q;MADP,e6HSI,M7HTJ,C;MACO,Q6HQH,  
M7HRG,+D;M6HSH,eAAQ,C;K;kGAIR,Y;MAAQ,0C;K;;ItK/KhB,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,  
8B;K;IAT3C,gF;sFAAA,yB;MAAA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;O;M  
AT3C,8E;MAAA,2B;QAQuF,2C;O;KARvF,C;IAiBgE,+C;MAAA,mB;QAAE,sB;O;K;IALIE,kC;MAKuD,OAAk  
B,2CAAT,+BAAS,E;K;IAEzE,8B;MAK6D,OAAI,QdksPtD,YAAQ,CclsP0C,GAAwB,eAAxB,GAA sD,WAAT,Q  
AAS,C;K;IAEnH,yB;MAG8C,kC;K;IAE9C,yB;MAAA,6B;K;uCACI,Y;MAA6C,kC;K;2CAC7C,a;MAA4B,kC;K;  
2CAC5B,a;MAA4B,kC;K;;IAHhC,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;oFAMA,yB;MAAA,2D;MAAA,4B;  
QAM4D,uCAAQ,e;O;KANpE,C;IAGB4F,mH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wD;MAAA,  
kC;K;;;kDAAA,Y;;;;;cACxF,eAAe,uBAAa,W;cAC5B,IAAI,QAAS,UAAb,C;gBACI,gB;gCAAA,sCAAS,QAAT,  
O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAEA,gB;gCAAA,sCAAS,iCAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;  
;;;;;cAJJ,W;;cAAA,W;;;;;;;K;IADwF,gE;MAAA,yD;uBAAA,uG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAP5  
F,4C;MAOmF,gBAAS,uCAAT,C;K;IAGBb,4B;MAAE,OAAA,EAAG,W;K;IAP3E,8B;MAO8D,4BAAQ,cAAR,C;  
K;IAUQ,8B;MAAE,OAAA,EAAG,W;K;IAR3E,8B;MAQ8D,4BAAQ,gBAAR,C;K;IAM1B,8B;MAAE,S;K;IAJtC,  
wC;MAEgB,Q;MADZ,IAAI,8CAAJ,C;QACI,OAA4C,CAApC,2EAAoC,kBAAQ,QAAR,C;;MAEHd,OAAO,uBA  
AmB,SAAnB,EAAyB,gBAazB,EAAiC,QAAjC,C;K;IAGX,4B;MAYiB,Q;MAFb,YAAY,gB;MACZ,YAAY,gB;M  
ACC,2B;MAAb,OAAa,cAAb,C;QAAs,sB;QACT,KAAM,WAAl,IAAK,MAAT,C;QACN,KAAM,WAAl,IAAK,O  
AAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;IAGX,+B;MAQqD,6BAAS,4BAAT,C;K;IAW0B,+G;MAAA,wC;MA  
AA,6B;MAAA,yB;MAAA,0C;MAAA,4C;MAAA,0B;MAAA,kC;K;;;mDAAA,Y;;;;;kCAC9D,0C;cACb,gB;;;;;cA  
AA,IAAO,iBT2FkD,US3FzD,C;gBAAA,gB;;;;;cACI,QAAQ,yBAAO,iBAAQ,iBAAO,KAAf,C;cACf,WAakB,WA  
AP,iBAAO,C;cACIB,YAAgB,IAAI,iBAAO,KAAf,GAAqB,iBAAO,aAAI,CAAJ,EAAO,IAAP,CAA5B,GAA8C,I;  
cAC1D,gB;8BAAA,iCAAM,KAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAJJ,gB;;;cAMJ,W;;;;;;;K;IAR+E,  
4D;MAAA,yD;uBAAA,mG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAT/E,uC;MASmE,gBAAY,kCAAZ,C;K;IAkB  
hC,0D;MAE/B,wB;QAAA,WAAgC,I;MADhC,0B;MACA,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;MACnC,gBA  
Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;QACI,WA  
AW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,MAAmB,sCAAvB,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ,M;;;  
MAGR,iBAAY,C;K;8DAGhB,Y;MASW,Q;MARP,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAj  
B,C;QACI,MAAM,6B;MACV,aAAa,a;MACb,gBAAW,I;MACX,iBAAY,E;MAEZ,OAAO,yE;K;iEAGX,Y;MACI,  
IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAhC5B,Y;MAAuC,yD;K;;IA2C3C,qD;MAAY,  
0B;MAAmC,gC;K;IACJ,gF;MAAA,0D;MACnC,gBA Ae,oCAAS,W;K;iEACxB,Y;MACI,OAAO,6CAAY,aAAS,O  
AArB,C;K;oEAGX,Y;MACI,OAAO,aAAS,U;K;;8CAPxB,Y;MAAuC,4D;K;qDAWvC,oB;MACI,OAAO,uBAA4B  
,eAA5B,EAA sC,kBAAtC,EAAmD,QAAnD,C;K;;IAUf,4D;MAAY,0B;MAAmC,gC;K;IACJ,8F;MAAA,wE;MAC  
nC,gBA Ae,2CAAS,W;MACxB,aAAY,C;K;wEACZ,Y;MAC0C,Q;MAAtC,OAAO,oDAAY,oBAAmB,iBAAnB,E  
AAmB,yBAAnB,QA AZ,EAAyC,aAAS,OAAID,C;K;2EAGX,Y;MACI,OAAO,aAAS,U;K;;qDARxB,Y;MAAuC,m  
E;K;;IAkB3C,oC;MAAY,0B;K;IAC6C,wE;MACJd,gBA Ae,gCAAS,W;MACxB,aAAY,C;K;6DACZ,Y;MAC2C,Q  
;MAAvC,OAAO,iBAAa,oBAAmB,iBAAnB,EAAmB,yBAAnB,QAAb,EAA0C,aAAS,OAA nD,C;K;gEAGX,Y;M  
ACI,OAAO,aAAS,U;K;;0CARxB,Y;MAAqD,wD;K;;IAMbZD,0D;MACI,4B;MACA,4B;MACA,4B;K;IAEuC,sE;  
MAAA,gD;MACnC,iBAAgB,gCAAU,W;MAC1B,iBAAgB,gCAAU,W;K;4DAC1B,Y;MACI,OAAO,sCAAU,cA  
AU,OAApB,EAA4B,cAAU,OAAtC,C;K;+DAGX,Y;MACI,OAAO,cAAU,UAAV,IAAuB,cAAU,U;K;;yCARhD,Y  
;MAAuC,uD;K;;IAc3C,6D;MACI,0B;MACA,gC;MACA,0B;K;IAEuC,4E;MAAA,sD;MACnC,gBA Ae,kCAAS,W;  
MACxB,oBAAiC,I;K;+DAEjC,Y;MACI,IAAI,CAAC,2BAAL,C;QACI,MAAM,6B;MACV,OAAO,gCA Ae,O;K;k  
EAG1B,Y;MACI,OAAO,2B;K;+EAGX,Y;MACQ,Q;MAAJ,IAAI,iEAA2B,KAA/B,C;QACI,oBAAe,I;MAEnB,O  
AAO,yBAAP,C;QACI,IAAI,CAAC,aAAS,UAAAd,C;UACI,OAAO,K;;UAEP,cAAc,aAAS,O;UACvB,uBAAuB,wC  
AAS,2CAAY,OAAZ,CAAT,C;UACvB,IAAI,gBAAiB,UAArB,C;YACI,oBAAe,gB;YACf,OAAO,I;;;MAInB,OA  
AO,I;K;;4CA9Bf,Y;MAAuC,0D;K;;IAoC9B,6I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,4C;MAAA,kD;MAAA  
,gD;MAAA,wB;MAAA,yB;MAAA,kC;K;;;yDAAA,Y;;;;;kBAGyC,I;iCAFIC,C;cACI,sD;cAAhB,gB;;;;;cAAA,KA  
AgB,yBAAhB,C;gBAAA,gB;;;;;cAAGB,oC;cACZ,aAAa,6BAAU,oBAAmB,uBAAnB,EAAmB,+BAAnB,QA AV,E



AAuC,OAAvC,C;cACb,gB;8BAAA,sCAAS,4BAAS,MAAT,CAAT,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB  
;;cAIJ,W;;;;;;K;IANS,0F;MAAA,yD;uBAAA,iI;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IADb,wD;MACI,gBA  
AS,kDAAT,C;K;;;IAoByB,qD;MACzB,0B;MACA,8B;MACA,0B;MC3TA,IAAI,ED+TQ,qBAAc,CC/TtB,CAAJ,  
C;QACI,cD8T2B,+CAA4C,iB;QC7TvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EDgUQ,mBAAy,CCh  
UpB,CAAJ,C;QACI,gBD+TyB,6CAA0C,e;QC9TnE,MAAM,gCAAYB,SAAQ,WAAjC,C;;MAFV,IAAI,EDiUQ,m  
BAAy,iBcJUpB,CAAJ,C;QACI,gBDgUkC,0DAAuD,eAAvD,WAAMe,iB;QC/TrG,MAAM,gCAAYB,SAAQ,WA  
AjC,C;;K;sFDkUa,Y;MAAQ,yBAAW,iBAAX,I;K;yCAE/B,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GA  
AqC,gBAAy,eAAZ,EAAsB,oBAAa,CAAb,IAAtB,EAAsC,eAAtC,C;K;yCAC9E,a;MAAyC,OAAI,KAAK,YAAT,  
GAAgB,IAAhB,GAA0B,gBAAy,eAAZ,EAAsB,iBAAtB,EAAkC,oBAAa,CAAb,IAAI,C;K;IAEzC,8D;MAAA,w  
C;MAEtB,gBAAe,2BAAS,W;MACxB,gBAAe,C;K;0DAEf,Y;MAEI,OAAO,gBAAW,kCAAX,IAAyB,aAAS,UAA  
zC,C;QACI,aAAS,O;QACT,qC;;K;2DAIR,Y;MACI,a;MACA,OAAQ,gBAAW,gCAAZ,IAAyB,aAAS,U;K;wDAG  
7C,Y;MACI,a;MACA,IAAI,iBAAy,gCAAhB,C;QACI,MAAM,6B;MACV,qC;MACA,OAAO,aAAS,O;K;;qCAvB  
xB,Y;MAA0B,mD;K;;IAgCA,uC;MAC1B,0B;MACA,oB;MC3WA,IAAI,ED+WQ,gBAAS,CC/WjB,CAAJ,C;QA  
CI,cD8WsB,yCAAsC,YAAtC,M;QC7WtB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgXV,a;MAAyC,OAAI,K  
AAK,YAAT,GAAgB,eAAhB,GAAqC,gBAAy,eAAZ,EAAsB,CAAtB,EAAyB,YAAzB,C;K;0CAC9E,a;MAAyC,  
OAAI,KAAK,YAAT,GAAgB,IAAhB,GAA0B,iBAAa,eAAb,EAAuB,CAAvB,C;K;IAE5B,gE;MACnC,YAAW,yB  
;MACX,gBAAe,4BAAS,W;K;yDAExB,Y;MACI,IAAI,cAAQ,CAAZ,C;QACI,MAAM,6B;MACV,6B;MACA,OA  
AO,aAAS,O;K;4DAGpB,Y;MACI,OAAO,YAAO,CAAP,IAAY,aAAS,U;K;;sCAZpC,Y;MAAuC,oD;K;;IASB3C,g  
D;MACI,0B;MACA,4B;K;IAEuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAA  
mB,I;K;oEAEnB,Y;MACI,IAAI,aAAS,UAb,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,CAAJ,C;U  
ACI,iBAAy,C;UACZ,gBAAW,I;UACX,M;;;MAGR,iBAAy,C;K;8DAGhB,Y;MAMiB,Q;MALb,IAAI,mBAAa,E  
AAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAJB,C;QACI,MAAM,6B;MACV,aACa,gF;MAGb,gBAAW,I;MACX,i  
BAAy,E;MACZ,OAAO,M;K;iEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2C  
AIC5B,Y;MAAuC,yD;K;;IA2Cb,uC;MAC1B,0B;MACA,oB;MC5bA,IAAI,ED+bQ,gBAAS,CC/bjB,CAAJ,C;QAC  
I,cD8bsB,yCAAsC,YAAtC,M;QC7btB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgcV,a;MItXO,SJsXmC,eAAQ  
,CAAR,I;MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,iBAAa,eAAb,EAAuB,EAAvB,C;K;0CACxG,a;  
MIVXO,SJuXmC,eAAQ,CAAR,I;MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,gBAAy,eAAZ,EAAsB,  
YAAtB,EAA6B,EAA7B,C;K;IAEjE,gE;MACnC,gBAAe,4BAAS,W;MACxB,YAAW,yB;K;2DAEX,Y;MAEI,OA  
AO,YAAO,CAAP,IAAY,aAAS,UAA5B,C;QACI,aAAS,O;QACT,6B;;K;yDAIR,Y;MACI,a;MACA,OAAO,aAAS,  
O;K;4DAGpB,Y;MACI,a;MACA,OAAO,aAAS,U;K;;sCAnBxB,Y;MAAuC,oD;K;;IA6B3C,gD;MACI,0B;MACA,  
4B;K;IAGuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;gEAEnB,Y;  
MACI,OAAO,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,CAAC,wCAAU,IAAV,CAAL,C;UACI,gB  
AAW,I;UACX,iBAAy,C;UACZ,M;;;MAGR,iBAAy,C;K;8DAGhB,Y;MAMqB,Q;MALjB,IAAI,mBAAa,EAAjB,  
C;QACI,a;MAEJ,IAAI,mBAAa,CAAJB,C;QACI,aACa,gF;QACb,gBAAW,I;QACX,iBAAy,C;QACZ,OAAO,M;;  
MAEX,OAAO,aAAS,O;K;iEAGpB,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,a;MACJ,OAAO,mBAAa,CAAb,IAA  
kB,aAAS,U;K;;2CAIC1C,Y;MAAuC,yD;K;;IAuCN,+C;MAAC,sB;MAAiC,gC;K;0CACnE,Y;MAAuC,4BAAiB,a  
AAO,WAAxB,EAAoC,kBAAP,C;K;;IAGP,+C;MAAuE,2B;MAAtE,sB;MAAiC,gC;MACIE,kBAAuB,c;K;6CAE  
vB,Y;MACI,OAAO,aAAO,UAAAd,C;QACI,WAAW,aAAO,O;QACIB,UAAU,mBAAy,IAAZ,C;QAEV,IAAI,eAA  
S,WAAI,GAAJ,CAAb,C;UACI,mBAAQ,IAAR,C;UACA,M;;;MAIR,W;K;;IAKgC,0D;MAAC,wC;MAAuC,kC;K;  
IACrC,0E;MAAA,oD;MACnC,gBAAmB,I;MACnB,iBAAqB,E;K;oEAERB,Y;MACI,gBAAe,mBAAa,EAAjB,GA  
AqB,+CAArB,GAA4C,2CAAa,4BAAb,C;MACvD,iBAAgB,qBAAJ,GAAsB,CAAtB,GAA6B,C;K;8DAG7C,Y;M  
AMiB,Q;MALb,IAAI,iBAAy,CAAhB,C;QACI,iB;MAEJ,IAAI,mBAAa,CAAJB,C;QACI,MAAM,6B;MACV,aAA  
a,8D;MAEb,iBAAy,E;MACZ,OAAO,M;K;iEAGX,Y;MACI,IAAI,iBAAy,CAAhB,C;QACI,iB;MACJ,OAAO,mB  
AAa,C;K;;2CAxB5B,Y;MAAuC,yD;K;;IA6B3C,kC;MAWI,OAAW,iDAAJ,GAAwC,SAAxC,GAakD,4BAAwB,S  
AAxB,C;K;IAeIB,uD;MAAA,qB;QAAE,6B;O;K;IAX7C,wC;MAWI,OAA2D,cAAPD,sBAakB,YAAIB,EAAgC,q  
CAAhC,CAAoD,C;K;IAqBrC,iD;MAAA,mB;QAAE,mB;O;K;IAIB5B,gD;MAeI,OAAI,YAAJ,GACI,2BADJ,GA  
GI,sBAakB,+BAAIB,EAA4B,YAA5B,C;K;IAER,wD;MAcI,6BAAkB,YAAIB,EAAgC,YAAhC,C;K;IPxpBJ,oB;  
MAAA,wB;MACI,8C;K;gCAEA,iB;MAA4C,oCAAmB,KAAM,U;K;kCACrE,Y;MAA+B,Q;K;kCAC/B,Y;MAAK

C,W;K;gFAEX,Y;MAAQ,Q;K;iCAC/B,Y;MAAkC,W;K;wCACIC,mB;MAAmD,Y;K;6CACnD,oB;MAAmE,OAA  
A,QAAS,U;K;kCAE5E,Y;MAA6C,kC;K;uCAE7C,Y;MAAiC,6B;K;;IADrC,gC;MAAA,+B;QAAA,c;;MAAA,wB;  
K;IAkBA,oB;MAIoC,6B;K;IAEpC,2B;MAMmD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,MAAT,QAAS,C  
AAhC,GAA6C,U;K;iFAEhG,yB;MAAA,mD;MAAA,mB;QAKwC,iB;O;KALx C,C;6FAOA,yB;MAAA,uE;MAAA  
,mB;QAQsD,2B;O;KARtD,C;IAUA,kC;MAKiE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAY,QAAS,OAArB,CAA  
d,CAAb,C;K;uFAE1E,yB;MAAA,2D;MAAA,mB;QAGgD,qB;O;KAHhD,C;IAKA,+B;MAC2D,OAAS,aAAT,QA  
AS,EAAa,eAAQ,YAAY,QAAS,OAArB,CAAR,CAAb,C;K;2FAEpE,yB;MAAA,uE;MAAA,mB;QAMwD,2B;O;K  
ANxD,C;IAQA,iC;MAKmE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAY,QAAS,OAArB,CAAd,CAAb,C;K;IAE5E,  
+B;MAMyD,OAAI,eAAJ,GAAqB,MAAM,OAAN,CAArB,GAAyC,U;K;IAElG,kC;MAQI,OAAgB,gBAAT,QAAS,  
EAAgB,sBAAhB,C;K;sFAGpB,yB;MetBA,uE;MfsBA,gC;QelB8B,gBAAnB,oB;QfqCiB,aWhDxB,W;QXgDA,O  
W/CO,SIUwC,Q;O;KfkBnD,C;wFA2BA,yB;Me1CA,wE;Mf0CA,0C;QetCsC,gBAA3B,mBf6DiB,Qe7DjB,C;Qf6D  
2B,aW/EIC,W;QX+EA,OW9EO,SliBgD,Q;O;KfsC3D,C;sFAGCA,yB;MAAA,mD;MAAA,4B;QAEkD,uCAAQ,U;  
O;KAF1D,C;IAIA,wC;MAAgD,QAAM,cAAN,C;aAC5C,C;UAD4C,OACvC,U;aACL,C;UAF4C,OAEvC,MAAM,  
oBAAW,OAAjB,C;;UAFuC,OAGpC,S;;K;IOrKZ,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;Iu  
KLA,yC;MtK4BI,IAAI,EsK3BI,OAAO,CAAP,IAAY,OAAO,CtK2BvB,CAAJ,C;QACI,csK3BI,aAAJ,GACI,yEA  
DJ,GAGI,8C;QtKyBJ,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;IsKnBM,mI;MAAA,mB;QAAE,wBAAiB,gCAAjB,  
EAA6B,YAA7B,EAAmC,YAAnc,EAAyC,sBAAzC,EAAyD,mBAAzD,C;O;K;IAFtB,gF;MACI,oBAAoB,IAApB,  
EAA0B,IAA1B,C;MACA,oCAAgB,6EAAhB,C;K;IAKyB,yL;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MA  
AA,wC;MAAA,gD;MAAA,sD;MAAA,4D;MAAA,wB;MAAA,0B;MAAA,uB;MAAA,0B;MAAA,wB;MAAA,qB;  
MAAA,4B;MAAA,kC;K;;;2DAAA,Y;;;;cACrB,4BAAiC,eAAL,uBAAK,EAAa,IAAb,C;+BACvB,0BAAO,uBAA  
P,I;cACV,IAAI,kBAAO,CAAX,C;oCACiB,iBAAa,qBAAb,C;kCACF,C;gBACD,6C;gBAAV,iB;;;sCAaa,gBAAc,q  
BAAd,C;gBACH,+C;gBAAV,gB;;;;;cAAA,KAAU,2BAAV,C;gBAAA,gB;;;cAAU,kC;cACN,mBAAO,WAAI,G  
AAJ,C;cACP,IAAI,mBAAO,SAAX,C;gBACI,IAAI,mBAAO,KAAP,GAAc,uBAAiB,C;kBAA0B,sBAAS,mBAAO  
,kBAAuB,uBAAvB,C;kBAA8B,gB;;;kBAAxE,gB;;;gBADJ,gB;;;cAGI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mB  
AAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C  
;cAJX,gB;;;cAFJ,gB;;;cASA,IAAI,iCAAJ,C;gBACI,gB;;;gBADJ,iB;;;cACI,IAAO,mBAAO,KAAAd,IAAqB,uBAA  
rB,C;gBAAA,gB;;;cACI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBA  
AA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C;cAFX,gB;;;cAIA,IhL8K4C,CgL9Kx C,mBhL8K  
yC,UgL9K7C,C;gBAAyB,iB;gCAAA,iCAAM,mBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAAzB,iB;;;cAj  
CR,W;;cA4BI,iB;;;cA1BJ,iB;;;cAGI,KAAU,yBAAV,C;gBAAA,iB;;;6BAAU,sB;cACN,IAAI,kBAAO,CAAX,C;g  
BAAgB,oCAAQ,CAAR,I;gBAAW,iB;;;gBAA3B,iB;;;cACA,iBAAO,WAAI,YAAJ,C;cACP,IAAI,iBAAO,KAAP  
,KAAe,uBAAAnB,C;gBACI,iB;gCAAA,iCAAM,iBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,iB;;;cAEI,  
IAAI,8BAAJ,C;gBAAiB,iBAAO,Q;;gBAAa,oBAAS,iBAAU,uBAAV,C;cAC9C,kBAAO,c;cAHX,iB;;;cAHJ,iB;;;c  
ASA,IhLiMgD,CgLjM5C,iBhLiM6C,UgLjMjD,C;gBACI,IAAI,qCAAkB,iBAAO,KAAP,KAAe,uBAArC,C;kBAA  
2C,iB;kCAAA,iCAAM,iBAAN,O;sBAAA,2C;2BAAA,yB;kBAAA,Q;;kBAA3C,iB;;;gBADJ,iB;;;cAdJ,W;;cAcI,  
iB;;;cAZJ,iB;;;cAkCJ,W;;;K;IARCyB,sI;MAAA,yD;uBAAA,6K;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAF  
7B,6E;MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,OAAO,2B;MACHC,OAAO,WAAkB,0EAAIB,C;K;IAwCwB,  
6B;MAA8B,uB;MAA7B,kB;MACHC,mBAA6B,C;MAC7B,eAAyB,C;K;2CAEzB,8B;MACI,+DAAkB,SAAlB,EA  
A6B,OAA7B,EAAcS,WAAK,KAA3C,C;MACA,mBAAiB,S;MACjB,eAAa,UAAU,SAAV,I;K;0CAGjB,iB;MACI  
,+DAAkB,KAAIB,EAAyB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAy,KAAZ,IAAL,C;K;qFAGY,Y;MAAQ,mB  
;K;;IASR,wC;MAAqD,uB;MAApD,sB;MtKrDxB,IAAI,EsKuDQ,cAAc,CtKvDtB,CAAJ,C;QACI,csKsD2B,wE;Qt  
KrD3B,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EsKwDQ,cAAc,aAAO,OtKxD7B,CAAJ,C;QACI,gBsK  
uDqC,wFAA+E,aAAO,O;QtKtD3H,MAAM,gCAAyB,SAAQ,WAAjC,C;;MsK2DV,kBAAuB,aAAO,O;MAC9B,o  
BAA8B,C;MAE9B,sBAAyB,U;K;kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;uCAGA,iB;MAGW,Q;MAF  
P,+DAAkB,KAAIB,EAAyB,SAAzB,C;MAEA,OAAO,sBAmGmC,CAnG5B,iBAmG6B,GAnGV,KAmGU,IAAD,I  
AAa,eAAb,IAngnC,4D;K;kCAGX,Y;MAAe,qBAAQ,e;K;IAEgB,4D;MAAA,sC;MAAS,2B;MAC5C,eAAoB,oB;  
MACpB,eAAoB,4B;K;8DAEpB,Y;MAKgB,Q;MAJZ,IAAI,iBAAS,CAAb,C;QACI,W;;QAGA,mBAAQ,sCAAQ,Y  
AAP,4DAAR,C;QACA,eAoFkC,CAPf1B,YAoF2B,GApFb,CAoFa,IAAD,IAAa,+BAAb,I;QAnFIC,mC;;K;;oCAX

Z, Y; MAAuC, kD; K; 2CAgBvC, iB; MAGiE, UAQ1C, MAR0C, EAe1C, MAf0C, EAqBtD, M; MATBP, aACQ, KAAM, O  
AAN, GAAa, IAAK, KAAtB, GAAkC, UAAN, KAAM, EAAO, IAAK, KAAZ, CAAIC, GAAyD, kD; MAE7D, WAAW, I  
AAK, K; MAEhB, WAAW, C; MACX, UAAU, iB; MAEV, OAAO, OAAO, IAAP, IA Ae, MAAM, eAA5B, C; QACI, OAA  
O, IAAP, IA Ae, wBAAO, GAAP, gE; QACf, mB; QACA, iB; MAGJ, MAAM, C; MACN, OAAO, OAAO, IAAd, C; QACI,  
OAAO, IAAP, IA Ae, wBAAO, GAAP, gE; QACf, mB; QACA, iB; MAEJ, IAAI, MAAO, OAAP, GAAc, IAAK, KAAvB, C  
; QAA6B, OAAO, IAAK, KAAZ, IA AoB, I; MAEjD, OAAO, uD; K; mCAGX, Y; MACI, OAAO, qBAAQ, gBAAa, SAAb,  
OAR, C; K; 4CAGX, uB; MAKI, kBAAoD, eAAjC, mBAAy, mBAAa, CAAzB, IAA8B, CAA9B, IAAiC, EAAa, WAAb,  
C; MACpD, gBAAoB, sBAAc, CAAIB, GAA4B, UAAP, aAAO, EAAO, WAAP, CAA5B, GAAqD, qBAAQ, gBAAa, WA  
Ab, OAR, C; MACrE, OAAO, eAAW, SAAX, EAA5B, SAAtB, C; K; qCAGX, mB; MAII, IAAI, aAAJ, C; QACI, MAAM,  
6BAAsB, qBAAtB, C; MAGV, cA6B0C, CA7BnC, iBA6BoC, GA7BjB, SA6BiB, IAAD, IA Aa, eAAb, IA7B1C, IAAmC,  
O; MACnC, 6B; K; +CAGJ, a; MtKhJA, IAAI, EsKoJQ, KAAK, CtKpJb, CAAJ, C; QACI, csKmJkB, wC; QtKIJb, MAAM,  
gCAAyB, OAAQ, WAAjC, C; MAFV, IAAI, EsKqJQ, KAAK, StKrJb, CAAJ, C; QACI, gBsKoJqB, wEAA8D, S; QtKnJn  
F, MAAM, gCAAyB, SAAQ, WAAjC, C; MsKqJN, IAAI, IAAI, CAAR, C; QACI, YAAy, iB; QACZ, UAgBsC, CAhB5B,  
KAgB6B, GAhBf, CAgBe, IAAD, IA Aa, eAAb, I; QAdtC, IAAI, QAAQ, GAAZ, C; UACW, OAAP, aAAO, EAAK, IAAL,  
EAAW, KAAX, EAAkB, eAAIB, C; UACA, OAAP, aAAO, EAAK, IAAL, EAAW, CAAX, EAAc, GAAd, C; UAEA, OA  
AP, aAAO, EAAK, IAAL, EAAW, KAAX, EAAkB, GAAIB, C; QAGX, oBAAa, G; QACb, wBAAQ, CAAR, I; K; qCAK  
R, wB; MAC8C, QAAC, YAAO, CAAP, IAAD, IA Aa, eAAb, I; K; IA9G9C, 0C; MAAA, oD; MAA6B, uBAAK, gBAAmB,  
QAA nB, OAAL, EAAMc, CAAnC, C; MAA7B, Y; K; ICvFJ, 0C; MAII, QAAQ, I; MACR, QAAQ, K; MACR, YAAy, kBA  
AM, CAAC, OAAO, KAAP, IAAD, IA AiB, CAAjB, IAAN, C; MACZ, OAAO, KAAK, CAAZ, C; QACI, OpL+B4E, 0BoL/  
BrE, kBAAM, CAAN, CpLoR2B, KAAL, GAAiB, GARP8B, EoL/B1D, KpLoRgB, KAAL, GAAiB, GARP8B, CoL/BrE, I  
AAP, C; UACI, a; QACJ, OpL6B4E, 0BoL7BrE, kBAAM, CAAN, CpLkR2B, KAAL, GAAiB, GARP8B, EoL7B1D, KpLk  
RgB, KAAL, GAAiB, GARP8B, CoL7BrE, IAAP, C; UACI, a; QACJ, IAAI, KAAK, CAAT, C; UACI, UAAU, kBAAM, CA  
AN, C; UACV, kBAAM, CAAN, EAAW, kBAAM, CAAN, CAAX, C; UACA, kBAAM, CAAN, EAAW, GAAX, C; UACA  
, a; UACA, a; MAGR, OAAO, C; K; IAGX, uC; MAGI, YAAy, aAAU, KAAV, EAAiB, IAAjB, EAAuB, KAAvB, C; MAC  
Z, IAAI, QAAO, QAAQ, CAAR, IAAP, CAAJ, C; QACI, UAAU, KAAV, EAAiB, IAAjB, EAAuB, QAAQ, CAAR, IAAvB  
, C; MACJ, IAAI, QAAQ, KAAZ, C; QACI, UAAU, KAAV, EAAiB, KAAjB, EAAwB, KAAxB, C; K; IAGR, 0C; MAII, QA  
AQ, I; MACR, QAAQ, K; MACR, YAAy, kBAAM, CAAC, OAAO, KAAP, IAAD, IA AiB, CAAjB, IAAN, C; MACZ, OA  
AO, KAAK, CAAZ, C; QACI, OILM6E, 0BkLNtE, kBAAM, CAAN, CILoP2B, KAAL, GAAiB, KA9O+B, EkLN3D, KIL  
oPgB, KAAL, GAAiB, KA9O+B, CkLNtE, IAAP, C; UACI, a; QACJ, OILi6E, 0BkLJtE, kBAAM, CAAN, CILkP2B, KAA  
L, GAAiB, KA9O+B, EkLJ3D, KILkPgB, KAAL, GAAiB, KA9O+B, CkLJtE, IAAP, C; UACI, a; QACJ, IAAI, KAAK, CA  
AT, C; UACI, UAAU, kBAAM, CAAN, C; UACV, kBAAM, CAAN, EAAW, kBAAM, CAAN, CAAX, C; UACA, kBAAM  
, CAAN, EAAW, GAAX, C; UACA, a; UACA, a; MAGR, OAAO, C; K; IAGX, yC; MAGI, YAAy, aAAU, KAAV, EAAiB,  
IAAjB, EAAuB, KAAvB, C; MACZ, IAAI, QAAO, QAAQ, CAAR, IAAP, CAAJ, C; QACI, YAAU, KAAV, EAAiB, IAAj  
B, EAAuB, QAAQ, CAAR, IAAvB, C; MACJ, IAAI, QAAQ, KAAZ, C; QACI, YAAU, KAAV, EAAiB, KAAjB, EAAwB,  
KAAxB, C; K; IAGR, 0C; MAII, QAAQ, I; MACR, QAAQ, K; MACR, YAAy, kBAAM, CAAC, OAAO, KAAP, IAAD, IA  
AiB, CAAjB, IAAN, C; MACZ, OAAO, KAAK, CAAZ, C; QACI, OnLnB8D, YmLmBvD, kBAAM, CAAN, CnLnBwE, K  
AAjB, EmLmB5C, KnLnByE, KAA7B, CmLmBvD, IAAP, C; UACI, a; QACJ, OnLrB8D, YmLqBvD, kBAAM, CAAN,  
CnLrBwE, KAAjB, EmLqB5C, KnLrByE, KAA7B, CmLqBvD, IAAP, C; UACI, a; QACJ, IAAI, KAAK, CAAT, C; UACI  
, UAAU, kBAAM, CAAN, C; UACV, kBAAM, CAAN, EAAW, kBAAM, CAAN, CAAX, C; UACA, kBAAM, CAAN, EA  
AW, GAAX, C; UACA, a; UACA, a; MAGR, OAAO, C; K; IAGX, yC; MAGI, YAAy, aAAU, KAAV, EAAiB, IAAjB, EA  
AuB, KAAvB, C; MACZ, IAAI, QAAO, QAAQ, CAAR, IAAP, CAAJ, C; QACI, YAAU, KAAV, EAAiB, IAAjB, EAAuB,  
QAAQ, CAAR, IAAvB, C; MACJ, IAAI, QAAQ, KAAZ, C; QACI, YAAU, KAAV, EAAiB, KAAjB, EAAwB, KAAxB, C;  
K; IAGR, 0C; MAII, QAAQ, I; MACR, QAAQ, K; MACR, YAAy, kBAAM, CAAC, OAAO, KAAP, IAAD, IA AiB, CAAjB  
, IAAN, C; MACZ, OAAO, KAAK, CAAZ, C; QACI, OIK5C+D, akK4CxD, kBAAM, CAAN, CIK5C0E, KAAIB, EkK4C7  
C, KIK5C2E, KAA9B, CkK4CxD, IAAP, C; UACI, a; QACJ, OIK9C+D, akK8CxD, kBAAM, CAAN, CIK9C0E, KAAIB,  
EkK8C7C, KIK9C2E, KAA9B, CkK8CxD, IAAP, C; UACI, a; QACJ, IAAI, KAAK, CAAT, C; UACI, UAAU, kBAAM, C  
AAN, C; UACV, kBAAM, CAAN, EAAW, kBAAM, CAAN, CAAX, C; UACA, kBAAM, CAAN, EAAW, GAAX, C; UAC  
A, a; UACA, a; MAGR, OAAO, C; K; IAGX, yC; MAGI, YAAy, aAAU, KAAV, EAAiB, IAAjB, EAAuB, KAAvB, C; MA

CZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAv  
B,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAKR,gD;MAI6E,  
UAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SA  
AjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAA  
V,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IrK9I7E,0C  
;MF0BI,IAAI,EEjBI,SAAU,OAAV,GAAiB,CFiBrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WA  
AjC,C;MEiBV,OAAO,oBAAoB,CAApB,EAAuB,CAAvB,EAA0B,SAA1B,C;K;IAGX,8C;MACe,Q;MAAX,wBA  
AW,SAAX,gB;QAAW,SAAS,SAAX,M;QACI,SAAS,GAAG,CAAH,C;QACT,SAAS,GAAG,CAAH,C;QACT,W  
AAW,cAAc,EAAc,EAAkB,EAAiB,C;QACX,IAAI,SAAQ,CAAZ,C;UAAe,OAAO,I;MAE1B,OAAO,C;K;sGAG  
X,yB;MAAA,8D;MAAA,iC;QASI,OAAO,cAAc,SAAS,CAAT,CAAd,EAA2B,SAAS,CAAT,CAA3B,C;O;KATX,  
C;sGAYA,sC;MASI,OAAO,UAAW,SAAQ,SAAS,CAAT,CAAR,EAAqB,SAAS,CAAT,CAArB,C;K;IAAtB,6B;M  
AWY,Q;MALR,IAAI,MAAM,CAAV,C;QAAa,OAAO,C;MACpB,IAAI,SAAJ,C;QAAe,OAAO,E;MACtB,IAAI,S  
AAJ,C;QAAe,OAAO,C;MAGtB,OAA8B,iBAAtB,mDAAsB,EAAU,CAAV,C;K;IAaZ,6C;MAAA,uB;QAAU,2BA  
AoB,CAApB,EAAuB,CAAvB,EAA0B,iBAA1B,C;O;K;IAVhC,8B;MF7CI,IAAI,EEsDI,SAAU,OAAV,GAAiB,CF  
tDrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;MEqDV,OAAO,eAAW,2BAAX,C;K;0  
FAIX,yB;MAAA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,  
OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MAPf,2B;QAOI,sB  
AAW,0BAAX,C;O;KAPJ,C;0FASA,yB;MAAA,oC;MAQe,gE;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;  
UAAIC,OA1Dd,UAAW,SAAQ,SA0DW,CA1DX,CAAR,EAAqB,SA0DC,CA1DD,CAArB,C;S;O;MAkDtB,uC;Q  
AQL,sBAAW,sCAAX,C;O;KARJ,C;4GAUA,yB;MAAA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;  
UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;  
W;S;OA+EI,C;MAPf,2B;QAOI,sBAAW,oCAAX,C;O;KAPJ,C;8GASA,yB;MAAA,oC;MAUe,0E;QAAA,uB;UAA  
U,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC,OA/Ed,UAAW,SAAQ,SA+EW,CA/EX,CAAR,EAAqB,SA+EC,CA/ED  
,CAArB,C;S;O;MAqEtB,uC;QAUl,sBAAW,gDAAX,C;O;KAVJ,C;kFAYA,yB;MAAA,sC;MAAA,oC;MAAA,oB  
AQe,yB;QA9Gf,8D;eA8Ge,yC;UAAA,uB;YACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;YACiC,Q;YA  
AA,IAAI,oBAAmB,CAAvB,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAvgG,cAAc,SAuG8C,CAvG9C,C  
AAd,EAA2B,SAuGoC,CAvGpC,CAA3B,C;;YAsGH,W;W;S;OADO,C;MARf,sC;QAQL,sBAAW,kCAAX,C;O;K  
ARJ,C;oFAaA,yB;MAAA,oC;MAQe,0E;QAAA,uB;UACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;UA  
CIC,Q;UAAA,IAAI,oBAAmB,CAAvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OA  
xGG,UAAW,SAAQ,SAwGyC,CaxGzC,CAAR,EAAqB,SAwG+B,CaxG/B,CAArB,C;;UAuGd,W;S;O;MATR,kD  
;QAQL,sBAAW,8CAAX,C;O;KARJ,C;sGAaA,yB;MAAA,sC;MAAA,oC;MAAA,8BAQe,yB;QAxIf,8D;eAwIe,m  
D;UAAA,uB;YACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;YAC5C,Q;YAAA,IAAI,oBAAmB,CAAv  
B,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAjIG,cAAc,SAiI8C,CAjI9C,CAAd,EAA2B,SAiIoC,CAjIpC,C  
AA3B,C;;YAgIH,W;W;S;OADO,C;MARf,sC;QAQL,sBAAW,4CAAX,C;O;KARJ,C;wGAaA,yB;MAAA,oC;MAQ  
e,8F;QAAA,uB;UACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;UAC5C,Q;UAAA,IAAI,oBAAmB,CA  
AvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OAIIG,UAAW,SAAQ,SAkIyC,CAIIZ  
C,CAAR,EAAqB,SAkI+B,CAII/B,CAArB,C;;UAiId,W;S;O;MATR,kD;QAQL,sBAAW,wDAAX,C;O;KARJ,C;kG  
AcA,yB;MAAA,oC;MAOe,wE;QAAA,uB;UACP,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;UAA1C,O  
ACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,mBAAW,CAAX,EAAc,CAAd,C;S;O;MATvD,wC;QAOI,sBAA  
W,4CAAX,C;O;KAPJ,C;IAmBe,oD;MAAA,uB;QACP,sBAAsB,SAAU,SAAQ,CAAR,EAAW,CAAX,C;QAAhC,  
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,uC;MAO  
IsBAAW,kCAAX,C;K;IAYc,wE;MAAA,uB;QACV,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;QAA1C,  
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,+C;MAO  
IsBAAc,4CAAd,C;K;IACw,+C;MAAA,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAeIB,E;aA  
Cb,c;UAHJ,OAGiB,C;;UAHjB,OAIY,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAb/B,gC;MAQL,sBAAW,6B  
AAX,C;K;4FASJ,yB;MAAA,4D;MAAA,wD;MAAA,mB;QAQqE,kBAAW,cAAAX,C;O;KARrE,C;IAkBe,8C;MAA  
A,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAeIB,C;aACb,c;UAHJ,OAGiB,E;;UAHjB,OAIY,  
kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAb/B,+B;MAQL,sBAAW,4BAAX,C;K;0FASJ,yB;MAAA,4D;MA

AA,sD;MAAA,mB;QAQoE,iBAAU,cAAV,C;O;KARpE,C;IAUA,wB;MAO4F,Q;MAA7B,OAA6B,4F;K;IAE5F,w  
B;MAO4F,Q;MAA7B,OAA6B,4F;K;IAE5F,gC;MAM+D,IAEJ,IAFI,EAGJ,M;MAFvD,kBAD2D,SAC3D,sB;QAD  
qD,OAC5B,SAAK,W;WAC9B,WAF2D,SAE3D,wC;QAFqD,OAE4F;WACvD,WAH2D,SAG3D,wC;QAHqD,O  
AGE,gG;;QAHF,OAI7C,uBAAmB,SAAnB,C;K;IAIuB,wC;MAAC,4B;K;2CACChC,gB;MAAwC,OAAA,eAAW,S  
AAQ,CAAR,EAAW,CAAX,C;K;4CACnD,Y;MACgC,sB;K;;IAGpC,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,  
iBAAF,CAAE,EAAU,CAAV,C;K;gDACE,Y;MAC8C,2C;K;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;IA  
MA,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACE,Y;MAC8C,2C;K;;I  
AHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;8EsKzTA,4B;MAUI,OAAK,iBAAL,SAAK,EAAU,KAAV,C;K;IC  
TT,iC;K;;;oDAyDI,0C;MAiB+D,oB;QAAA,2C;aAjB/D,kG;K;;IAoBJ,uC;MAAA,e;MAAA,iB;MAAA,uB;K;IAA  
A,qC;MAAA,wC;O;MASI,4E;MAMA,8E;MAOA,4E;MAOA,kE;K;;IApBA,mD;MAAA,2B;MAAA,2C;K;;IAMA,  
oD;MAAA,2B;MAAA,4C;K;;IAOA,mD;MAAA,2B;MAAA,2C;K;;IAOA,8C;MAAA,2B;MAAA,sC;K;;IA7BJ,iC;  
MAAA,+K;K;;IAAA,sC;MAAA,a;aAAA,c;UAAA,gD;aAAA,e;UAAA,iD;aAAA,c;UAAA,gD;aAAA,S;UAAA,2C  
;;UAAA,oE;;K;;oFAqCA,mB;K;;;IhImBiD,gD;MAAA,oB;QACzC,WAAW,sBAAmB,YAAF,CAAE,C  
AAnB,C;QACX,cAAM,IAAN,C;QADA,OAEA,IAAK,a;O;K;;;IAtHb,+B;K;;iFAUA,yB;MAAA,4B;MAAA,mC;  
QAMI,6BDgDQ,WChDkB,KDgDIB,CChDR,C;O;KANJ,C;2GAQA,yB;MAAA,4B;MDgDQ,kD;MChDR,uC;QAO  
I,6BDgDQ,WAAO,cChDW,SDgDX,CAAP,CChDR,C;O;KAPJ,C;+FAUA,yB;MAAA,kC;MAAA,mD;MAAA,yE;  
QASI,sC;QAAA,4C;O;MATJ,iGAWY,Y;QAAQ,2B;OAXpB,E;MAAA,0DAaQ,kB;QACI,wBAAW,MAAX,C;O;  
MAZ,sF;MAAA,sC;QASI,0D;O;KATJ,C;IAiBA,gD;MAaI,4BAA0D,YAAzC,wCAA6B,UAA7B,CAAyC,CAA1  
D,EAAyE,yBAzE,C;K;IAEJ,4D;MAcI,4BAAoE,YAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAApE,  
EAAmF,yBAAnF,C;K;IAEJ,+C;MAU6C,YAAzC,wCAA6B,UAA7B,CAAyC,CAtEzC,oBDgDQ,WCSBsD,kBDtB  
tD,CChDR,C;K;IAyEJ,2D;MAWuD,YAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAPFnD,oBDgDQ,WC  
oCgE,kBDpChE,CChDR,C;K;IAuFJ,+C;MAYI,OAA6C,8BAAtC,c;K;8EAZX,yB;MAAA,oE;MAAA,6E;MAYiD,  
gD;QAAA,oB;UACzC,WAAW,sBAAmB,YAAF,CAAE,CAAnB,C;UACX,cAAM,IAAN,C;UADA,OAEA,IAAK,  
a;S;O;MAfb,sC;QAYW,mBAAsC,8BAAtC,6B;QAAP,OAAO,kD;O;KAZX,C;qGA0BI,yB;MAAA,2D;MAAA,mB  
;QACI,MAAM,6BAAoB,0BAApB,C;O;KADV,C;;MiIzIA,yC;;IAAA,uC;MAAA,2C;K;;IAAA,mD;MAAA,kD;Q  
AAA,iC;;MAAA,2C;K;+EakBA,wB;K;oDAaA,e;MAK2C,IAAI,IAAJ,EAGK,M;MAL5C,IAAI,+CAAJ,C;QAEI,O  
AAW,GAAL,kBAAS,IAAK,IAAd,CAAR,GAA4B,cAAI,OAAJ,GAAL,iBAAQ,IAAR,CAAJ,yCAA5B,GAAyD,I;;  
MAGpE,OAAW,8CAA4B,GAAhC,GAAqC,8EAArC,GAAoD,I;K;yDAI/D,e;MAGI,IAAI,+CAAJ,C;QACI,OAA  
W,GAAL,kBAAS,IAAK,IAAd,CAAJ,IAA0B,GAAL,iBAAQ,IAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,I;;  
MAE/F,OAAW,8CAA4B,GAAhC,GAAqC,mCAArC,GAAgE,I;K;;;ICtChD,oD;MACf,cAAc,GAAL,kBAAS,OAA  
Q,IAAjB,C;MACIB,IAAI,YAAY,mCAAhB,C;QADA,OACuC,O;;QAEEnC,kBAAkB,oBAAQ,yCAAR,C;QACIB,I  
AAI,mBAAJ,C;UAJJ,OAI6B,oBAAgB,OAAhB,EAAyB,OAAzB,C;;UACrB,WAAW,OAAQ,kBAAS,yCAAT,C;U  
AL3B,OAMY,SAAS,mCAAb,GAAoC,oBAAgB,OAAhB,EAAyB,WAAzB,CAApC,GACI,oBAAgB,oBAAgB,IA  
AhB,EAAzB,OAAtB,CAAhB,EAAgD,WAAhD,C;;K;8CAdxB,mB;MAKI,OAAI,YAAY,mCAAhB,GAAuC,IAAV  
C,GACI,OAAQ,cAAK,IAAL,EAAW,4BAAX,C;K;;;qDAiCZ,e;MAEyB,Q;MADrB,OACI,OAAA,IAAK,IAAL,  
EAAy,GAAZ,CAAJ,GAAqB,0EAARB,GAAoC,I;K;sDAExC,8B;MACI,iBAAU,OAAV,EAAmB,IAAnB,C;K;0D  
AEJ,e;MACI,OAAI,OAAA,IAAK,IAAL,EAAy,GAAZ,CAAJ,GAAqB,mCAArB,GAAgD,I;K;;;IC1DP,8C;MAAC  
,wB;K;kFAAA,Y;MAAA,yB;K;;IAiCe,wD;MAEjE,kC;MAEA,4BAAqC,mDAAJ,GAakD,OAAQ,qBAA1D,GAA  
0E,O;K;4DAE3G,mB;MAA6C,+BAAS,OAAT,C;K;6DAC7C,e;MAA8C,eAAQ,IAAR,IAAgB,8BAAe,G;K;;IAGjF  
,+C;MAW2C,IAAI,IAAJ,EAGV,M;MAL7B,IAAI,+CAAJ,C;QAEI,OAAW,GAAL,kBAAS,SAAK,IAAd,CAAR,G  
AA4B,cAAI,OAAJ,GAAL,iBAAQ,SAAR,CAAJ,yCAA5B,GAAyD,I;;MAGpE,OAAW,SAAK,IAAL,KAAa,GAAj  
B,GAAsB,mFAAtB,GAAqC,I;K;IAGhD,6C;MAUI,IAAI,+CAAJ,C;QACI,OAAW,GAAL,kBAAS,SAAK,IAAd,C  
AAJ,IAA0B,GAAL,iBAAQ,SAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,S;;MAE/F,OAAW,SAAK,IAAL,K  
AAa,GAAjB,GAAsB,mCAAtB,GAAiD,S;K;IAG5D,iC;MAAA,qC;MAKI,4B;K;oDACA,Y;MAAiC,0C;K;kDAEj  
C,e;MAAyD,W;K;mDACzD,8B;MAA4E,c;K;mDAC5E,mB;MAAwE,c;K;uDACxE,e;MAA8D,W;K;+CAC9D,Y;  
MAAsC,Q;K;+CACtC,Y;MAAyC,8B;K;;IAB7C,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAqB8B,wC;MAC1B,k  
B;MACA,wB;K;4CAGA,e;MAGQ,Q;MAFJ,UAAU,I;MACV,OAAO,IAAP,C;QACI,YAAA,GAAL,UAAJ,aAAY,  
GAAZ,W;UAAwB,W;;QACxB,WAAW,GAAL,O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,iBAAK,

GAAL,C;;;K;6CAKnB,8B;MACI,iBAAU,WAAK,cAAK,OAAL,EAAc,SAAd,CAAf,EAAyC,cAAzC,C;K;iDAEJ,e  
;UAGW,I;MAFP,+BAAQ,GAAR,U;QAAoB,OAAO,W;;;MAC3B,cAAc,WAAK,kBAAS,GAAT,C;MAEf,gBAAY,  
WAAZ,C;QAAoB,W;WACpB,gBAAY,mCAAZ,C;QAAqC,qB;;QAC7B,2BAAgB,OAaHb,EAAyB,cAAzB,C;MA  
HZ,W;K;uCAOJ,Y;MAIc,IAAI,IAAJ,Q;MAHV,UAAU,I;MACV,WAAW,C;MACX,OAAO,IAAP,C;QACU,uBA  
AI,OAAJ,GAAI,OAAJ,gC;QAAA,mB;UAAgC,OAAO,I;;QAA7C,MAAM,M;QACN,mB;;K;2CAIR,mB;MACI,+  
BAAI,OAAQ,IAAZ,GAAoB,OAApB,C;K;8CAEJ,mB;MAQ4B,Q;MAPxB,UAAU,O;MACV,OAAO,IAAP,C;QA  
CI,IAAI,CAAC,gBAAS,GAAI,UAAb,CAAL,C;UAA4B,OAAO,K;QACnC,WAAW,GAAI,O;QACf,IAAI,oCAAJ,  
C;UACI,MAAM,I;;UAEN,OAAO,gBAAS,0EAAT,C;;;K;uCAKnB,iB;MACI,gBAAS,KAAT,KAakB,yCAA4B,K  
AAM,SAAN,KAAgB,aAA5C,IAAsD,KAAM,eAAY,IAAZ,CAA9E,C;K;yCAEJ,Y;MAA+B,OAAK,SAAL,WAA  
K,CAAL,GAA0B,SAAR,cAAQ,CAA1B,I;K;IAGZ,uD;MACX,OAAI,GzJyHoC,YAAU,CyJzHID,GAAMb,OAAQ  
,WAA3B,GAA6C,GAAF,UAAQ,O;K;yCAF3D,Y;MACI,aAAM,kBAAK,EAAL,EAAS,+BAAT,CAAN,GAEL,G;  
K;IAMO,8E;MAAA,6B;QAAyB,Q;QAAT,iBAAS,sBAAT,EAAS,8BAAT,UAAoB,O;QAAQ,W;O;K;+CAJ3D,Y;  
MAOsB,Q;MANIB,QAAQ,a;MACR,eAAe,gBAA+B,CAA/B,O;MACf,gBAAY,CAAZ,C;MACA,kBAAK,kBAAL  
,EAAW,oDAAX,C;M5KtFJ,IAAI,E4KuFM,YAAS,C5KvFf,CAAJ,C;QACI,cAdW,e;QaEx,MAAM,6BAAsB,OA  
AQ,WAA9B,C;;M4KuFN,OAAO,+BAAW,qDAAX,C;K;IAGa,8C;MACpB,kD;MADqB,wB;K;IACrB,gD;MAAA,  
oD;MACI,4B;K;;;IADJ,4D;MAAA,2D;QAAA,0C;;MAAA,oD;K;yDAIA,Y;MAA0C,gBAAT,a;M3L8/YrB,Q;MA  
DhB,kB2L7/YmD,mC;M3L8/YnD,wBAAgB,SAaHb,gB;QAAgB,cAAA,SAaHb,M;QAAsB,cAAwB,yBAAa,OA  
Ab,C;;M2L9/YT,O3L+/Y9B,W;K;;;I4LjrZX,oE;MA4BI,MAAM,wBAAoB,sEAAPB,C;K;8GA5BV,yB;MAAA,2D  
;MAAA,sC;QA4BI,MAAM,6BAAoB,sEAAPB,C;O;KA5BV,C;IA0CoC,mC;MAAQ,4D;K;IAE5C,4C;MAAA,e;M  
AAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAK0C,oG;MAAqB,gF;MAAW,4E;K;;IAAhC,+D;MAAA,gC;M  
AAA,uD;K;;IAAqB,qD;MAAA,gC;MAAA,6C;K;;IAAW,mD;MAAA,gC;MAAA,2C;K;;IAL1E,sC;MAAA,sJ;K;;I  
AAA,2C;MAAA,a;aAAA,qB;UAAA,4D;aAAA,W;UAAA,kD;aAAA,S;UAAA,gD;;UAAA,qF;;K;;;ICrCa,sC;M  
AG0F,2BAAgB,eAAhB,C;K;IAKE,6C;MAAA,mB;QAAE,sB;O;K;IAH9F,gC;MAGwG,gBAA5B,oBAAgB,2BAA  
hB,C;M3KoEIE,S2K1DH,K;MAVqE,O3KqEjE,S;K;I2KzB+B,0C;MAAgE,uB;MAA/D,wC;MAMvC,kBACkC,I;K  
;4FAE9B,Y;MACI,QAAQ,e;MACR,IAAI,SAAJ,C;QAAe,OAAO,C;MACtB,IAAI,wB;MACJ,kBAAW,C;MACX,  
OAAO,C;K;uFAIX,Y;MAAQ,OAAA,cAAQ,O;K;4CAEPb,iB;MACI,cAAc,c;MACd,+DAAKB,KAAlB,EAAyB,O  
AAQ,OAAjC,C;MACA,OAAO,QAAQ,KAAR,C;K;+CAMX,mB;MAEI,IAAI,YAAY,IAAhB,C;QAAsB,OAAO,K  
;MAE7B,aAAqB,UAAAR,cAAQ,EAAU,OAAQ,QAAlB,C;MACrB,OAAO,WAAW,O;K;8CAGtB,mB;MAEI,IAAI,  
YAAY,IAAhB,C;QAAsB,OAAO,E;MAE7B,cAAc,OAAQ,Q;MACtB,aAAqB,UAAAR,cAAQ,EAAU,OAAV,C;MA  
CrB,OAAW,WAAW,OAAf,GAAwB,OAAxB,GAAqC,E;K;kDAGhD,mB;MAA4C,0BAAQ,OAAR,C;K;+CAE5C,  
Y;MAEI,OAAO,kCAA8B,cAA9B,C;K;;ICpHf,gC;K;;ICAA,sC;K;;6ECAA,yB;MAAA,0B;MAAA,mC;QAGsD,O  
AAiC,OAA3B,SAAL,GAAuB,KAAS,C;O;KAHvF,C;2EAKA,yB;MAAA,0B;MAAA,mC;QAGqD,OAAgC,OAA1  
B,SAAL,GAAsB,KAAS,C;O;KAHrF,C;6EAKA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA3B,SAAL,GAA  
uB,KAAS,C;O;KAHvF,C;6EAKA,yB;MAAA,0B;MAAA,4B;QAGqC,OAAqB,OAAP,CAAR,SAAE,C;O;KAH1D,  
C;+EAMA,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;6EAKA,y  
B;MAAA,4B;MAAA,mC;QAGwD,OAAgC,QAA1B,SAAL,GAAsB,KAAS,C;O;KAHxF,C;+EAKA,yB;MAAA,4  
B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;+EAKA,yB;MAAA,4B;MAAA,4B  
;QAGuC,OAAqB,QAAP,CAAR,SAAE,C;O;KAH5D,C;ICpCA,qC;K;;ICAA,mB;K;;IAOA,iB;K;;IAOA,2C;K;;IAO  
A,wB;K;;IAQA,0B;K;;IAOA,sB;K;;IAOA,4B;K;;IAOA,6C;K;;IA+BuC,wE;MAEnC,uB;QAAA,UAAAsB,E;MACt  
B,qB;QAAA,8B;MACA,2B;QAAA,qE;MACA,yB;QAAA,YAAqB,E;MAJrB,sB;MACA,sB;MACA,kB;MACA,8B  
;MACA,0B;K;;IAGJ,iD;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,+C;MAAA,kD;O;MAKI,wG;MACA,wG;MAC  
A,8F;K;;IAFA,iE;MAAA,qC;MAAA,yD;K;;IACA,iE;MAAA,qC;MAAA,yD;K;;IACA,4D;MAAA,qC;MAAA,oD;  
K;;IAPJ,2C;MAAA,6K;K;;IAAA,gD;MAAA,a;aAAA,kB;UAAA,8D;aAAA,kB;UAAA,8D;aAAA,a;UAAA,yD;;U  
AAA,6E;;K;;IAUA,wB;K;;ICnGA,mB;MAEI,UAAU,IAAI,CAAJ,I;MACV,OAAW,OAAO,CAAX,GAAC,GAAD,  
GAAuB,MAAM,CAAN,I;K;IAGiC,qB;MACI,UAAU,SAAI,CAAJ,C;MACV,OAAW,kBAAO,CAAX,GAAC,GAAD,  
GAAuB,QAAM,CAAN,C;K;IAGiC,mC;MAEI,OAAO,IAAI,IAAI,CAAJ,EAAO,CAAP,IAAY,IAAI,CAAJ,EAA  
O,CAAP,CAAZ,IAAJ,EAA2B,CAA3B,C;K;IAGX,qC;MACI,OAAO,MAAI,MAAI,CAAJ,EAAO,CAAP,WAAZ,  
MAAI,CAAJ,EAAO,CAAP,CAAZ,CAAJ,EAA2B,CAA3B,C;K;IAGX,qD;MAkBI,WAAO,CAAP,C;QAD2E,OAC

3D,SAAS,GAAb,GAaKB,GAAIB,GAA2B,MAAM,iBAaiB,GAAjB,EAA6B,IAA7B,CAAN,I;W  
ACvC,WAAO,CAAP,C;QAF2E,OAE3D,SAAS,GAAb,GAaKB,GAAIB,GAA2B,MAAM,iBAaiB,KAajB,EAAw  
B,GAAxB,EAA6B,CAAC,IAAD,IAA7B,CAAN,I;;QAC/B,MAAa,gCAAYB,eAAzB,C;K;IAGzB,uD;MAkBI,sBA  
AO,CAAP,C;QAD+E,OAC/D,sBAAS,GAAT,MAAJ,GAaKB,GAAIB,GAA2B,aAAM,mBAaiB,GAAjB,EAA6B,  
KAAtB,EAA6B,IAA7B,CAAN,C;WACvC,sBAAO,CAAP,C;QAF+E,OAE/D,sBAAS,GAAT,MAAJ,GAaKB,GA  
AIB,GAA2B,QAAM,mBAaiB,KAajB,EAAwB,GAAxB,EAA8B,IAAD,aAA7B,CAAN,C;;QAC/B,MAAa,gCAA  
yB,eAAzB,C;K;IC7DzB,qB;MAAA,yB;K;OCAII,Y;MAO6D,uB;K;2HAE7D,yB;MAAA,+D;MAAA,kC;MAAA,0  
F;MAAA,6F;MAAA,4E;QAUI,wC;QAAS,2C;O;MAVb,mEAWQ,wC;QAA6E,sBAAS,QAAT,EAAmB,QAAnB,E  
AA6B,QAA7B,C;O;MAXrF,oG;MAAA,yC;QAUI,wDAA+B,YAA/B,C;O;KAVJ,C;uHAcA,yB;MAAA,+D;MAA  
A,kC;MAAA,wF;MAAA,yF;MAAA,0E;QAcI,wC;QAAS,2C;O;MADB,kEAeQ,wC;QAAuF,6BAAS,QAAT,EAAm  
B,QAAnB,EAA6B,QAA7B,C;O;MAf/F,kG;MAAA,yC;QAcI,sDAA+B,YAA/B,C;O;KADJ,C;;;IA3BJ,iC;MAAA,g  
C;QAAA,e;;MAAA,yB;K;IAGDiC,sB;MAC7B,eAAwB,I;K;4CAExB,6B;MACoB,IAAT,I;MAAA,mB;MAAA,iB;  
QAAS,MAAM,6BAASB,cAAy,QAAS,aAArB,uCAAtB,C;;MAAtB,OAAO,I;K;4CAGX,oC;MACI,eAAa,K;K;;;k  
DC9CjB,6B;;K;;;;iEA+CA,6B;;K;;ICrDuC,0C;MACvC,uBAaOB,Y;K;wDAEPB,wC;MAM6F,W;K;uDAE7F,w  
C;K;oDAMA,6B;MACI,OAAO,oB;K;oDAGX,oC;MACI,eAAe,IAAK,gB;MACpB,IAAI,CAAC,0BAAa,QAAb,E  
AAuB,QAavB,EAAiC,KAajC,CAAL,C;QACI,M;;MAEJ,uBAaA,K;MACb,yBAAY,QAaz,EAA6B,QAAT,EAA  
gC,KAaHc,C;K;;4EC9BR,wC;MAqBI,OAAO,e;K;4EAGX,+C;MAuBI,cAAI,KAaj,C;K;4EAIJ,wC;MAmBI,OA  
AO,cAAI,OAAJ,C;K;4EAGX,+C;MAqBI,cAAI,OAAJ,EAAa,KAAb,C;K;IC/FJ,kB;MA6PI,4B;K;+BAtoA,Y;MA  
OiC,6BAAS,EAAT,C;K;uCAEjC,iB;MAW2C,4BAAQ,CAAR,EAaw,KAAX,C;K;uCAE3C,uB;MAakB,Q;MAHd  
,iBAaiB,IAajB,EAAuB,KAavB,C;MACA,QAAQ,QAAQ,IAAR,I;MACR,IAAI,IAAI,CAAJ,IAAS,MAAK,WAA  
IB,C;QACc,IAAI,MAAM,CAAC,CAAD,IAAN,OAAY,CAaHb,C;UACN,eAAe,SAAS,CAAT,C;UACf,6BAAS,Q  
AAT,C;;UAEA,K;;YAEI,WAAW,cAAU,KAak,C;YAC1B,IAAI,OAAO,CAAP,I;;UACC,gBAAO,CAAP,IAAY,C  
AAZ,GAAGB,CAaHb,SAAqB,CAArB,C;UACT,Q;;QATJ,c;QAWA,OAAO,OAAO,GAAP,I;;QAEP,OAAO,IAAP  
,C;UACI,YAAU,c;UACV,IAAW,IAAP,qBAakB,KAAtB,C;YAA6B,OAAO,K;;K;gCAKhD,Y;MAOmC,OAAU,o  
BAAV,cAAU,CAAS,WAAI,EAaj,CAAnB,yBAA6B,cAA7B,E;K;wCAEnC,iB;MAW8C,iCAAY,KAaz,C;K;wC  
AE9C,uB;MAiBkB,Q;MAPd,mBAaiB,IAajB,EAAuB,KAavB,C;MACA,QAAQ,eAAQ,IAAR,C;MACR,IAAI,eA  
AI,CAAR,C;QACI,O;QACA,IAAI,aAAO,CAAD,aAAN,GAAY,CAAZ,CAAJ,C;UACI,WAAW,CAAE,Q;UACb,  
YAAa,qBAAO,EAAP,CAAW,Q;UAEpB,aAAQ,CAAR,C;YACI,eAAe,SAAS,IAAT,C;YAEf,OAAmB,oBAAnB,s  
BAAS,QAAT,CAAmB,CAAnB,iB;iBAEJ,cAAS,CAAT,C;YAEI,OAAU,oBAAV,cAAU,CAAV,iB;;YAEA,iBAAe  
,SAAS,KAAT,C;YACf,OAAmB,oBAAnB,sBAAS,UAAT,CAAmB,CAAS,WAAI,EAaj,CAA5B,KAaiD,oBAAV  
,cAAU,CAAV,iBAavC,C;;UAXR,U;;UAeA,K;;YAEI,WAAW,eAAW,oBAAK,CAAL,C;YACTB,IAAI,YAAO,CA  
AP,C;;UACC,sBAAO,CAAP,MAAY,+BAAI,CAAJ,EAaz,eAAqB,CAArB,C;UACT,MAAM,C;;QAEV,OAAO,S  
AAO,GAAP,C;;QAEP,OAAO,IAAP,C;UACI,YAAU,e;UACV,IAAW,IAAP,0CAakB,KAaIB,CAAJ,C;YAA6B,O  
AAO,K;;K;mCAKhD,Y;MAKyC,6BAAS,CAAT,MAAe,C;K;kCAExD,Y;MAKuC,uBAAGB,sBAAS,EAAT,CAA  
hB,EAA8B,sBAAS,EAAT,CAA9B,C;K;0CAEvC,iB;MASoD,+BAAW,GAAX,EAAGB,KAaHb,C;K;0CAEPD,uB;  
MAcY,Q;MAFR,mBAaiB,IAajB,EAAuB,KAavB,C;MACA,WAAW,QAAQ,I;MACX,IAAS,WAAI,IAAK,CAA  
L,IAA0B,SAAL,IAAK,CAA1B,IAA8C,SAAN,KAAM,CAAID,C;QACJ,SAAS,qBAAGB,QAAQ,CAAR,GAAY,O  
AAO,CAAnC,C;QACT,cAAO,EAAP,GAAY,E;;QAEZ,cAAO,oBAAE,I;;MAJ1B,Y;MAMA,OAAW,KAak,KA  
T,GAASB,SAAN,KAAM,CAAtB,GAAS,C,C;K;iCAGjD,Y;MAKqC,6BAAS,EAAT,IAA0B,Q;K;IAWK,oF;MAAA  
,mB;QAAE,uBAaA,iBAAb,sBAAqC,eAArC,+BAAqE,aAAM,OAA3E,M;O;K;iDATtE,qC;MzLjLA,IAAI,EyL0Lq  
B,CAAb,8BAAGB,KAAM,OzL1L9B,GyL0LiD,CAAX,0BAAc,KAAM,OzL1L1D,GyL0LsC,KzL1LiC,CAAJ,C;Q  
ACI,cyLyLgE,kDzLzLID,E;QACd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EyL2LQ,aAAa,OzL3LrB,C  
AAJ,C;QACI,gByL0LgC,mF;QzLzLhC,MAAM,gCAAYB,SAAQ,WAAjC,C;;MyL2LN,YAAY,CAAC,UAAU,SA  
AV,IAAD,IAAwB,CAAxB,I;MAEZ,mBAAe,SAaf,C;MtLzEJ,iBAAc,CAAd,UsL0EW,KtL1EX,U;QsL2EQ,QAA  
Q,c;QACR,MAAM,UAAN,IAAoB,OAAF,CAAE,C;QACpB,MAAM,aAAW,CAAX,IAAN,IAAGC,OAAV,CAAE,  
KAak,CAAG,C;QACc,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAx,CAAE,KAak,EAAL,C;QACjC,MAAM,a  
AAW,CAAX,IAAN,IAAiC,OAAx,CAAE,KAak,EAAL,C;QACjC,0BAAy,CAAZ,I;;MAGJ,gBAAGB,UAAU,UA  
AV,I;MACHB,SAAS,sBAAS,YAAY,CAAZ,IAAT,C;MACT,aAAU,CAAV,MAakB,SAaIB,M;QACI,MAAM,aA

AW,CAAX,IAAN,IAAqC,OAAf,EAAG,MAAK,IAAI,CAAJ,IAAL,CAAY,C;;MAGzC,OAAO,K;K;yCACX,uD;  
MAvB4C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,O;aARrF,0H;K;yCAiCA,iB;MAOyD,8BAAU,K  
AAV,EAAiB,CAAjB,EAAoB,KAAM,OAA1B,C;K;yCAEzD,gB;MAKkD,8BAAU,cAAU,IAAV,CAAV,C;K;IAGI  
D,0B;MAAA,8B;MAO2B,iB;MACvB,uBAAoC,uB;K;IAEpC,qC;MAAA,yC;MACI,4B;K;wDAEA,Y;MAAiC,mC  
;K;;;IAHrC,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;8CAMA,Y;MAAkC,8C;K;gDAElC,oB;MAA4C,OAAA,oBA  
Ac,kBAAS,QAAT,C;K;uCAC1D,Y;MAA8B,OAAA,oBAAc,U;K;+CAC5C,iB;MAAwC,OAAA,oBAAc,iBAAQ,K  
AAR,C;K;+CACtD,uB;MAAmD,OAAA,oBAAc,iBAAQ,IAAR,EAAC,KAAc,C;K;wCAEjE,Y;MAAgC,OAAA,oB  
AAc,W;K;gDAC9C,iB;MAA2C,OAAA,oBAAc,kBAAS,KAAT,C;K;gDACzD,uB;MAAuD,OAAA,oBAAc,kBAA  
S,IAAT,EAAe,KAAf,C;K;2CAErE,Y;MAAsC,OAAA,oBAAc,c;K;0CAEpD,Y;MAAoC,OAAA,oBAAc,a;K;kDA  
CID,iB;MAAiD,OAAA,oBAAc,oBAAW,KAAX,C;K;kDAC/D,uB;MAA+D,OAAA,oBAAc,oBAAW,IAAX,EAAi  
B,KAAjB,C;K;yCAE7E,Y;MAAkC,OAAA,oBAAc,Y;K;iDAEhD,iB;MAAsD,OAAA,oBAAc,mBAAU,KAAV,C;  
K;iDACpE,gB;MAA+C,OAAA,oBAAc,mBAAU,IAAV,C;K;yDAC7D,qC;MACI,OAAA,oBAAc,mBAAU,KAAV  
,EAAiB,SAAjB,EAA4B,OAA5B,C;K;;;IAtCtB,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IA0CJ,wB;MAuC,yBA  
Aa,IAAb,EAAb,IAAK,IAAI,EAA5B,C;K;IAEvC,wB;MAawC,yBAAa,IAAK,QAAIB,EAA2B,IAAK,YAAI,EA  
AJ,CAAQ,QAAxC,C;K;IAGxC,mC;MAUI,IAAA,KAAM,UAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D  
,C;WACzB,IAAA,KAAM,KAAN,GAAa,UAAb,C;QAF8C,OAEhB,0BAAQ,KAAM,MAAd,EAAqB,KAAM,KAA  
N,GAAa,CAAb,IAArB,C;WAC9B,IAAA,KAAM,MAAN,GAAC,WAAc,C;QAH8C,OAGf,0BAAQ,KAAM,MAA  
N,GAAc,CAAd,IAAR,EAAYB,KAAM,KAA/B,IAAuC,CAAvC,I;;QAHe,OAIc,mB;K;IAGZ,oC;MAUI,IAAA,K  
AAM,UAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KAAN,+C;QAFiD,OAEI  
B,2BAAS,KAAM,MAAf,EAAsB,KAAM,KAAN,yBAAa,CAAb,EAAtB,C;WAC/B,IAAA,KAAM,MAAN,+C;QA  
HiD,OAGjB,2BAAS,KAAM,MAAN,8BAAC,CAAd,EAAT,EAA0B,KAAM,KAAhC,0BAAwC,CAAxC,E;;QAHi  
B,OAIzC,oB;K;IAOZ,yB;MAAyC,YrFrTbB,YqFqTb,KrFrTa,CqFqTb,I;K;IAEzC,4C;MAEI,OAAA,SAAK,KAA  
K,EAAL,GAAU,QAAf,GAAyC,CAAX,CAAC,QAAD,IAAW,KAAI,E;K;IAEjD,uC;MzLTVI,IAAI,EyLsVuD,QA  
AQ,IzLTV/D,CAAJ,C;QACI,cyLqVuE,+B;QzLpVvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IyLqVd,yC;MzLVI  
IAAI,EyLuVyD,sBAAQ,IAAR,KzLvVzD,CAAJ,C;QACI,cyLsVyE,+B;QzLrVzE,MAAM,gCAAYB,OAAQ,WAAj  
C,C;;K;IyLsVd,yC;MzLxVI,IAAI,EyLwV6D,QAAQ,IzLxVrE,CAAJ,C;QACI,cyLuV6E,+B;QzLTV7E,MAAM,gC  
AAyB,OAAQ,WAAjC,C;;K;IyLwVd,yC;MAAyD,oCAA0B,IAA1B,qBAAiC,KAAjC,kB;K;ICrXzD,6B;MAOqC,  
OtM6YE,SsM7YF,mBtM6YE,C;K;IsM3YvC,sC;MASgD,6BAAS,WAAAT,EAAa,KAAb,C;K;IAEhD,4C;MAUI,qB  
AAqB,IAArB,EAA2B,KAA3B,C;MAEA,iBAAiB,ItM+QgB,KsM/QhB,GAAiB,W;MACiC,kBAAkB,KtM8Qe,Ks  
M9Qf,GAAkB,W;MAEpC,mBAAMB,0BAAQ,UAAAR,EAAoB,WAApB,IAAqC,W;MACxD,OtMgXmC,SsMhX5  
B,YtMgX4B,C;K;IsM7WvC,sC;MAWI,IAAA,KAAM,UAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C;;  
QACzB,ItMGkE,YsMHIE,KAAM,KtMG6E,KAAjB,EsMhRd,4BAAK,UtMG6E,KAA7B,CsMHIE,K;UAFiD,OAE  
IB,sBAAS,KAAM,MAAf,EtMqBsB,SsMrBA,KAAM,KtMqBI,KAAK,GAAW,CsMrBb,WtMqBa,MAAX,IAAf,Cs  
MrBtB,C;;UAC/B,ItMEKE,YsMFIE,KAAM,MtME6E,KAAjB,EsMFpD,4BAAK,UtME4E,KAA7B,CsMFIE,K;YA  
HiD,OtMuBI,SsMpBrB,sBtMiCsB,SsMjCb,KAAM,MtMiCiB,KAAK,GAAW,CsMjC1B,WtMiC0B,MAAZ,IAAf,C  
sMjCtB,EAA2B,KAAM,KAAjC,CtMoB+B,KAAK,GAAW,CsMpBN,WtMoBM,MAAX,IAAf,C;;YsMvBJ,OAIzC  
,mB;;K;IAGZ,8B;MAOuC,OrLoWG,UqLpWH,oBrLoWG,C;K;IqLIW1C,uC;MASmD,8BAAU,2BAAV,EAAe,K  
AAf,C;K;IAEnD,6C;MAUI,sBAAsB,IAAtB,EAA4B,KAA5B,C;MAEA,iBAAiB,IrLkOkB,KqLlOIB,8B;MACjB,k  
BAAkB,KrLiOiB,KqLjOjB,8B;MAEIB,mBAAMB,2BAAS,UAAAT,EAAqB,WAArB,+B;MACnB,OrLuUsC,UqLv  
U/B,YrLuU+B,C;K;IqLpU1C,uC;MAWI,IAAA,KAAM,UAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C  
;;QACzB,IrL7CmE,aqL6CnE,KAAM,KrL7C+E,KAAIB,EqL6CtD,6BAAM,UrL7C8E,KAA9B,CqL6CnE,K;UAFo  
D,OAEpB,uBAAU,KAAM,MAAhB,ErLhCuB,UqLgCA,KAAM,KrLhCK,KAAK,KAAW,CjBgR7C,UAAW,oBA  
AL,CsMhPyB,WtMgPzB,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,CqLgCvB,C;;UACHC,IrL9CmE,aqL8  
CnE,KAAM,MrL9C+E,KAAIB,EqL8CrD,6BAAM,UrL9C6E,KAA9B,CqL8CnE,K;YAHoD,OrL9BG,UqLiCtB,uB  
rLpBuB,UqLoBb,KAAM,MrLpBkB,KAAK,UAAAY,CjBmQ/C,UAAW,oBAAL,CsM/Oc,WtM+Od,MAAK,CAAL,i  
BAAN,CiBnQ+C,MAAZ,CAAhB,CqLoBvB,EAA4B,KAAM,KAAIC,CrLjCiC,KAAK,KAAW,CjBgR7C,UAAW,  
oBAAL,CsM/OgC,WtM+OhC,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,C;;YqL8BH,OAI5C,oB;;K;IAG  
Z,sC;MAQI,4BAAU,KIKg/FH,QkKh/FP,C;MACA,OAAO,K;K;IAGX,uC;MAKsD,OIK+iG3C,ekK/iG2C,4BAAU,



IAAV,CIK+iG3C,C;K;IkK7iGX,4D;MAOgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,K;MACrF,4  
BAAU,KIK69FH,QkK79FP,EAA+B,SAA/B,EAA0C,OAA1C,C;MACA,OAAO,K;K;IAIX,2C;M1LrHI,IAAI,EZ2  
B8D,YsM0FD,KtM1FkB,KAAjB,EsM0FO,ItM1FsB,KAA7B,CsM0FD,I1LrH7D,CAAJ,C;QACI,c0LoH6E,+B;Q1  
LnH7E,MAAM,gCAAYB,OAAQ,WAAjC,C;K;I0LoHd,4C;M1LrHI,IAAI,EKmC+D,aqLmFC,KrLnFiB,KAAIB,E  
qLmFS,IrLnFqB,KAA9B,CqLmFC,I1LtHhE,CAAJ,C;QACI,c0LqHgF,+B;Q1LpHhF,MAAM,gCAAYB,OAAQ,W  
AAjC,C;K;I2LpBc,6C;MAScxB,oC;MA/BA,iB;MANA,Y;MACA,Y;MACA,Y;MACA,Y;MACA,Y;MACA,sB;M  
3LYA,IAAI,E2LLQ,CAAC,WAAK,QAAL,GAAU,QAAY,GAAe,QAaf,GAAoB,QAARB,MAA2B,C3LKnC,CAA  
J,C;QACI,c2LNwC,wD;Q3LOxC,MAAM,gCAAYB,OAAQ,WAAjC,C;MGoHV,iBAAc,CAAd,UwLxHW,ExLwH  
X,U;QwLxHiB,c;K;qCAGjB,Y;MAGI,QAAQ,Q;MACR,IAAI,IAAO,MAAO,C;MACIB,WAAI,Q;MACJ,WAAI,  
Q;MACJ,WAAI,Q;MACJ,SAAS,Q;MACT,WAAI,E;MACJ,IAAK,IAAO,KAAM,CAAd,GAAsB,EAAtB,GAA8B,  
MAAO,C;MACzC,WAAI,C;MACJ,gCAAU,MAAV,I;MACA,OAAO,IAAI,aAAJ,I;K;8CAGX,oB;MACI,OAAU,c  
AAV,cAAU,EAAC,QAAd,C;K;IAEd,kC;MAAA,sC;MACI,4B;K;IADJ,8C;MAAA,6C;QAAA,4B;MAAA,sC;K;  
IA7BA,gD;MAAA,sD;MACQ,yBAAK,KAAL,EAAY,KA AZ,EAAMB,CAAnB,EAAsB,CAAtB,EAA+B,CAAN,K  
AAzB,EAAuB,CAAU,EAAX,GAAoB,UAAW,CAArE,C;MADR,Y;K;IIMbkB,wC;MA8BIB,iC;MA9BsD,2BAAG  
B,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFAC7B,Y;MAAQ,8B;K;yFACD,Y;MAAQ,6B;K;yFAKR,Y;M  
AC5B,IAAI,cAAQ,sCAAK,UAAjB,C;QOyHyC,MAAM,6BPzHb,6EOyH2C,WAA9B,C;MPxH/C,OAAO,+BAA  
O,CAAP,E;K;2CAGX,iB;MAA8C,qBAAS,KAAT,IAAkB,SAAS,S;K;kCAEzE,Y;MAKkC,oBAAQ,S;K;iCAE1C,i  
B;MACI,2CAAuB,kBAaA,KAAM,UAAAnB,KACvB,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KADf,CAAvB,  
C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,OAAK,UsBUS,ItBVd,UAAkB,SsBUJ,ItBVd,K;K;mCA  
E5B,Y;MAAKC,2BAAE,UAAF,+BAAU,SAAV,C;K;IAEIC,+B;MAAA,mC;MACI,aAC8B,cAAAY,OAAF,CAAE,C  
AAZ,EAAwB,OAAF,CAAE,CAAxB,C;K;IAFIC,2C;MAAA,0C;QAAA,yB;MAAA,mC;K;IAUiB,uC;MA8BjB,  
gC;MA9BmD,0BAAe,KAAf,EAAsB,YAAtB,EAAoC,CAAP,C,C;K;iFAC3B,Y;MAAQ,iB;K;wFACD,Y;MAAQ,g  
B;K;wFAKR,Y;MAC3B,IAAI,cAAQ,UAAZ,C;QOIFyC,MAAM,6BPjFd,6EOIF4C,WAA9B,C;MPHF/C,OAAO,Y  
AAO,CAAP,I;K;0CAGX,iB;MAA6C,qBAAS,KAAT,IAAkB,SAAS,S;K;iCAExE,Y;MAKkC,oBAAQ,S;K;gCAE1  
C,iB;MACI,0CAAsB,kBAaA,KAAM,UAAAnB,KACtB,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KADhB,CAA  
tB,C;K;kCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,MAAK,UAAAL,QAAa,SAAb,I;K;kCAE5B,Y;MAAK  
C,OAAE,UAAF,qBAAU,S;K;IAE5C,8B;MAAA,kC;MACI,aAC6B,aAAS,CAAT,EAAY,CAAZ,C;K;IAFjC,0C;  
MAAA,yC;QAAA,wB;MAAA,kC;K;IAUkB,wC;MA8BIB,iC;MA9BsD,2BAAGB,KAAhB,EAAuB,YAAvB,K;K;  
kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;yFAKR,Y;MAC5B,IAAI,2CAAJ,C;QOyCyC,MAAM,6BPzCb,  
6EOyC2C,WAA9B,C;MPxC/C,OAAO,kCAAQ,CAAP,E;K;2CAGX,iB;MAA8C,kCAAS,KAAT,UAAkB,sBAAS,  
SAAT,M;K;kCAEhE,Y;MAKkC,kCAAQ,SAAR,K;K;iCAEIC,iB;MACI,2CAAuB,kBAaA,KAAM,UAAAnB,KACv  
B,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CADD,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,G  
AAe,EAaf,GAAwB,iCAAM,eAAW,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EAAY,CAAV,CAA  
pC,CAA8D,Q;K;mCAE1F,Y;MAAKC,OAAE,UAAF,qBAAU,SAAV,W;K;IAEIC,+B;MAAA,mC;MACI,aAC8B,q  
B;K;IAFIC,2C;MAAA,0C;QAAA,yB;MAAA,mC;K;ImM9GkC,oD;MAA2C,uB;MAAjB,gB;MAC5D,sBAAGC,  
I7KmCU,I;M6KIC1C,iBAAmC,YAAO,CAAX,GAAc,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAA5B,GA  
AqC,K7KiCK,I6KjC1C,GAAqD,mB;K;gDAErD,Y;MAAKC,qB;K;iDAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,UAA  
S,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;QAGV,4BAAQ,SAAR,I;MAEJ,O  
AAa,OAAAN,KAAM,C;K;IAQgB,mD;MAAYC,sB;MAAjB,gB;MACzD,sBAAGC,I;MACHC,iBAAmC,YAAO,CA  
AX,GAAc,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAAJ,GAAa,KAAb,GAAwB,mB;K;+CAEhD,Y;MAA  
kC,qB;K;+CAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,UAAAS,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,  
6B;QAC3B,iBAAU,K;QAGV,4BAAQ,SAAR,I;MAEJ,OAAO,K;K;IAQuB,oD;MAA4C,uB;MAAIB,gB;MAC5D  
,sBAAiC,I;MACjC,iBAAmC,uBAAO,CAAX,GAAc,sBAAS,IAAT,MAAd,GAAiC,sBAAS,IAAT,M;MACHC,cAA  
6B,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAEjD,Y;MAAKC,qB;K;iDAEIC,Y;MACI,YAAAY,W;MACZ,IAAI,cAA  
S,mBAAT,CAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;QAGV,8BAAQ,SAAR,C;  
MAEJ,OAAO,K;K;IC9DX,oD;MA6CA,uC;MATCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAYB,wBAAzB,C;M  
AC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAYB,wEAAzB,C;MAG5C,aAGyB,K;MAEZB,YAGuF,OAA/D,  
0BAA0B,K9KeR,I8KfIB,EAAsC,Y9KepB,I8KfIB,EAAYD,IAAZD,CAA+D,C;MAEvF,YAGuB,I;K;yCAEvB,Y;M

AAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,aAAQ,SAATB,GAAgC,aAAQ,S;K;uCAE7E,iB;MACI,iDAA6B,kBAaA,KAAM,UAAAnB,KAC7B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAiC,IAA8C,cAAQ,KAAM,KAD/B,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,OAAK,U9KPG,I8KOR,UAAkB,S9KPV,I8KOR,KAAN,SAAqC,SAArC,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAc,oBAAE,UAAF,+BAAU,SAAV,eAAqB,SAAnC,GAA8C,oBAAE,UAAF,qCAAgB,SAAhB,gBAA4B,CAAC,SAAD,IAA5B,C;K;IAEhF,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,UAAhB,EAA4B,QAA5B,EAA5C,IAAtC,C;K;;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAiBA,mD;MA6CA,sC;MAiCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAYB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAYB,wEAAzB,C;MAG5C,aAGwB,K;MAExB,YAGuB,0BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEvB,YAGuB,I;K;wCAEvB,Y;MAAuC,kCAAuB,UAAvB,EAA8B,SAA9B,EAAoC,SAAPC,C;K;uCAEvC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,aAAQ,SAATB,GAAgC,aAAQ,S;K;sCAE7E,iB;MACI,gDAA4B,kBAaA,KAAM,UAAAnB,KAC5B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAiC,IAA8C,cAAQ,KAAM,KADhC,CAA5B,C;K;wCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UAAI,QAAa,SAAb,IAAN,SA A2B,SAA3B,I;K;wCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GA AgD,UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEhF,oC;MAAA,wC;K;iEACI,sC;MAQwF,0B AAe,UAAf,EAA2B,QAA3B,EAAqC,IAArC,C;K;;;IAT5F,gD;MAAA,+C;QAAA,8B;;MAAA,wC;K;;IAiBA,oD;M A6CA,uC;MAiCI,IAAI,gBAAJ,C;QAAgB,MAAa,gCAAYB,wBAAzB,C;MAC7B,IAAI,sCAAJ,C;QAA4B,MAAa, gCAAYB,yEAAzB,C;MAG7C,aAGyB,K;MAEzB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C; MAExB,YAGwB,I;K;yCAExB,Y;MAAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y; MAMqC,OAAI,uBAAO,CAAX,GAAc,2BAAQ,SAAR,KAAd,GAAgC,2BAAQ,SAAR,K;K;uCAErE,iB;MACI,iD AA6B,kBAaA,KAAM,UAAAnB,KAC7B,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CAAxB,IAA8C ,kBAAQ,KAAM,KAAd,CADjB,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,iCAA M,eAAW,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EA AV,CAAV,CAAPC,CAAN,MAAuE,cAAU, 6BAAU,EA AV,CAAV,CAA vE,CAAiG,Q;K;yCAE7H,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU ,SAAV,yBAAqB,SAArB,WAA d,GAAgD,UAAF,2BAAgB,SAAhB,yBAA6B,SAAD,aAA5B,W;K;IAEhF,qC;MA AA,yC;K;kEACI,sC;MAQ4F,2BAAgB,UAAhB,EAA4B,QAA5B,EAA5C,IAAtC,C;K;;;IAThG,iD;MAAA,gD;QA AA,+B;;MAAA,yC;K;;;6CCIKa,iB;MAKkD,+BAAS,UAAT,UAAkB,wBAAS,iBAAT,M;K;oCAEpE,Y;MAKgC, oCAAQ,iBAAR,K;K;;;8CAuBhC,iB;MAKkD,+BAAS,UAAT,UAAkB,wBAAQ,iBAAR,K;K;qCAEpE,Y;MAKgC ,oCAAS,iBAAT,M;K;;ICxDiB,8C;MACjD,4B;MACA,0C;K;oEADA,Y;MAAA,2B;K;2EACA,Y;MAAA,kC;K;uC AGA,iB;MACI,OAAO,0CAAgC,kBAaA,KAAM,UAAAnB,KAC/B,mBAAS,KAAM,MAAf,KAAwB,0BAAgB,KA AM,aAAtB,CADo,CAAhC,C;K;yCAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN,UAAAM,CA AX,QAAqC,SAAb,iBAaA,CAArC,I;K;yCAGIC,Y;MAAkC,OAAE,UAAF,qBAAU,iB;K;;IAGhD,kC;MAM6E,2B AAgB,SAAhB,EAA5B,IAAtB,C;K;IAMjB,qD;MACxD,4B;MACA,0C;K;2EADA,Y;MAAA,2B;K;kFACA,Y;MA AA,kC;K;8CAGA,iB;MACI,OAAO,iDAAuC,kBAaA,KAAM,UAAAnB,KACtC,mBAAS,KAAM,MAAf,KAAwB,0 BAAgB,KAAM,aAAtB,CADc,CAA vC,C;K;gDAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN, UAAM,CAAX,QAAqC,SAAb,iBAaA,CAArC,I;K;gDAGIC,Y;MAAkC,OAAE,UAAF,sBAAW,iB;K;;IAGjD,qC; MAQiF,kCAAuB,SAAvB,EAA6B,IAA7B,C;K;;;0DAY7E,iB;MAA2C,qCAAiB,UAAjB,EAAwB,KAAxB,KAAk C,8BAAiB,KAAjB,EAAwB,iBAAxB,C;K;iDAC7E,Y;MAAkC,QAAC,8BAAiB,UAAjB,EAAwB,iBAAxB,C;K;;I AcR,gD;MAI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0FACD,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;2DAE1C,g B;MAA+D,YAAK,C;K;mDAEpE,iB;MAAgD,gBAAS,aAAT,IAAmB,SAAS,oB;K;0CAC5E,Y;MAAkC,SAAE,iB AAU,oBAAZ,C;K;yCAEIC,iB;MACI,OAAO,4CAA+B,kBAaA,KAAM,UAAAnB,KAC9B,kBAAU,KAAM,SAAhB ,IAA0B,yBAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SA AP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAA vC,I;K;2CAGIC,Y;MAAkC,OAAE,aAAF,qBAAW,oB;K;;IAGjD,o C;MAOqF,6BAAkB,SAAlB,EAAwB,IAAxB,C;K;IAQRd,iD;MAI5B,gBAAqB,K;MACrB,uBAA4B,Y;K;2FACD, Y;MAAQ,oB;K;kGACD,Y;MAAQ,2B;K;sDAE1C,gB;MAA8D,YAAK,C;K;oDAEnE,iB;MAAgD,gBAAS,aAAT,I AAmB,QAAQ,oB;K;2CAC3E,Y;MAAkC,SAAE,gBAAS,oBAAX,C;K;0CAEIC,iB;MACI,OAAO,6CAAgC,kBAa a,KAAM,UAAAnB,KAC/B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADiB,CAAhC,C;K;4CAIX,Y;MA CI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAA vC,I;K;4CAGIC,

Y;MAAkC,OAAE,aAAF,sBAAY,oB;K;;IAGID,uC;MAO4E,8BAAMb,SAAnB,EAAyB,IAAZB,C;K;IAQ9C,+C;M  
AI1B,gBAAqB,K;MACrB,uBAA4B,Y;K;yFACF,Y;MAAQ,oB;K;gGACD,Y;MAAQ,2B;K;0DAEzC,gB;MAA6D,  
YAAK,C;K;kDAEIE,iB;MAA+C,gBAAS,aAAT,IAAMb,SAAS,oB;K;yCAC3E,Y;MAAkC,SAAE,iBAAU,oBAA  
Z,C;K;wCAEIC,iB;MACI,OAAO,2CAA8B,kBAaA,KAAM,UAAAnB,KAC7B,kBAAU,KAAM,SAAhB,IAA0B,yB  
AAiB,KAAM,gBADpB,CAA9B,C;K;0CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,  
CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;0CAGIC,Y;MAAkC,OAAE,aAAF,qBAAW,oB;K;;IAGjD,oC;MAOkF,  
4BAAiB,SAAjB,EAAuB,IAAvB,C;K;IASnD,gD;MAI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0FACF,Y;MAAQ,o  
B;K;iGACD,Y;MAAQ,2B;K;qDAEzC,gB;MAA4D,YAAK,C;K;mDAEjE,iB;MAA+C,gBAAS,aAAT,IAAMb,QA  
AQ,oB;K;0CAC1E,Y;MAAkC,SAAE,gBAAS,oBAAX,C;K;yCAEIC,iB;MACI,OAAO,4CAA+B,kBAaA,KAAM,  
UAAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX,Y;MACI,OAA  
W,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;2CAGIC,Y;MAA  
kC,OAAE,aAAF,sBAAY,oB;K;;IAGID,uC;MAOyE,6BAakB,SAAlB,EAawB,IAAxB,C;K;oFAGzE,8B;MAQI,0  
BAAMb,2BAAS,OAAT,C;K;oFAEvB,8B;MASI,0BAAMb,2BAAS,OAAT,C;K;IAEvB,+C;MACI,IAAI,CAAC,U  
AAL,C;QAAiB,MAAM,gCAAyB,iCAA8B,IAA9B,iBAAZB,C;K;ICxQ3B,gC;MAcW,Q;MADP,IAAI,CAAC,6BA  
AW,KAAX,CAAL,C;QAAwB,MAAM,uBAAMb,sCtFjBzC,oBsFiByC,CAAnB,C;;MAC9B,OAAO,sD;K;IAMX,o  
C;MAakC,Q;MAA9B,OAAW,6BAAW,KAAX,CAAJ,GAAuB,sDAAvB,GAAuC,I;K;;;ICvBhB,yC;MA2B9B,u  
C;MA1BA,wB;MAIA,gB;MjMQA,IAAI,EiMDS,iBAAY,IAAb,MAAuB,iBAAvB,CjMCR,CAAJ,C;QACI,ciMDQ,  
iBAAY,IAAhB,GACI,8CADJ,GAGI,sCAA0B,aAA1B,qC;QjMDR,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;yCiM  
KV,Y;MAAwC,Q;MAAA,oB;MACpC,iB;QAD8B,OAcTb,G;WACR,oD;QAF8B,OAEF,SAAL,SAAK,C;WAC5B,  
6C;QAH8B,OAGd,iBAAK,SAAL,C;WACHb,8C;QAJ8B,OAIb,kBAAM,SAAN,C;;QAJa,mC;K;IAOIC,qC;MAAA  
,yC;MACI,YAGqC,oBAAgB,IAAhB,EAAsB,IAAtB,C;K;iGAQJ,Y;MAAQ,gB;K;4DAEzC,gB;MAOI,8DAAqC,I  
AArC,C;K;gEAEJ,gB;MAMI,uDAA8B,IAA9B,C;K;4DAEJ,gB;MAMI,wDAA+B,IAA/B,C;K;;;IArCR,iD;MAAA,  
gD;QAAA,+B;;MAAA,yC;K;;2CArCJ,Y;MAWI,oB;K;2CAXJ,Y;MAeI,gB;K;6CafJ,0B;MAAA,2BAWI,8CAXJ,E  
AeI,kCafJ,C;K;yCAAA,Y;MAAA,c;MAWI,yD;MAIA,qD;MAfJ,a;K;uCAAA,iB;MAAA,4IAWI,4CAXJ,IAeI,oC  
AfJ,I;K;ICLA,kC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,gC;MAAA,mC;O;MAYI,4D;MAKA,8C;MAKA,gD;K  
;;IAVA,2C;MAAA,sB;MAAA,mC;K;;IAKA,oC;MAAA,sB;MAAA,4B;K;;IAKA,qC;MAAA,sB;MAAA,6B;K;;IA  
tBJ,4B;MAAA,mG;K;;IAAA,iC;MAAA,a;aAAA,W;UAAA,wC;aAAA,I;UAAA,iC;aAAA,K;UAAA,kC;;UAAA,6  
D;;K;;6ECAA,yB;MAAA,4F;MAAA,2B;QASI,MAAM,mCAA8B,0EAA9B,C;O;KATV,C;ICkCA,+D;MAAw,Q;  
MAAP,OAAO,8CAA0,KAAP,EAAC,UAAAd,EAA0B,QAA1B,oC;K;IAGX,kC;MAIiB,Q;MAAb,wBAAa,KAAb,g  
B;QAAA,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;mFAGX,qB;MAGwD,gCAA0,EAAP,C;K  
;qFAExD,4B;MAG4E,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAOXD,4B;MAGmE,OAAA,yBAAO  
,KAAP,CAVX,gBAAO,EAAP,C;K;IAaxD,wD;MAEQ,sB;QAAqB,yBAAO,UAAU,OAAV,CAAP,C;WACrB,sD;  
QAA4B,yBAAO,OAAP,C;WAC5B,2B;QAAMb,yBAAO,kBAAP,C;;QACX,yBAAe,SAAR,OAAQ,CAAf,C;K;ItL  
7EhB,+B;MAY6B,kBAAIB,QAAQ,SAAR,EAAC,EAAd,C;MACH,IX0EE,WW1EE,GAAK,CAAT,C;QAAy,MA  
AM,gCAAyB,oEAzB,C;MADtB,OX4EO,W;K;IWvEX,wC;MAGBsC,IAA3B,I;MAAA,qCAAiB,KAAjB,C;MAA  
A,iB;QAA2B,MAAM,gCAAyB,8BAAO,SAAP,4CAA+C,KAAXE,C;;MAAXC,OAAO,I;K;IAGX,qC;MAY6B,kB  
AAIB,QAAQ,SAAR,EAAC,EAAd,C;MAAP,OXmEqB,WWnEa,IAAM,CXmEjC,GAAqB,WAArB,GAA+B,I;K;I  
WhE1C,8C;MAGBI,WAAW,KAAX,C;MAC4B,kBAArB,QAAQ,SAAR,EAAC,KAAd,C;MAAP,OX+CqB,WW/Cg  
B,IAAM,CX+CpC,GAAqB,WAArB,GAA+B,I;K;IW5C1C,gC;MAWI,IAAY,CAAR,8BAAW,CAAf,C;QACI,OA  
AO,YAAM,SAAN,C;;MAEX,MAAM,gCAAyB,SAAM,SAAN,4BAAZB,C;K;IAGV,yC;MAkBW,Q;MANP,IAAI,  
EAAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAyB,oBAAiB,KAAjB,4CAAzB,C;;MAEV,IAAI,YAA  
O,CAAP,IAAY,aAAQ,KAAXB,C;QACI,MAAM,gCAAyB,WAAQ,SAAR,mDAAwD,KAAjF,C;;MAEH,IAAI,YA  
AO,EAAX,C;QACH,mBAAM,SAAN,C;;QAEA,0BAAM,SAAN,IAAa,EAAb,C;;MAHJ,W;K;IAuFJ,8B;MAWsC,  
+B;K;0EAEtC,4B;MAM8D,OAAK,oBAAL,SAAK,CAAL,GAakB,K;K;IAEHf,gD;MAQoC,0B;QAAA,aAAsB,K;  
MACtD,IAAI,cAAQ,KAAX,C;QAAMb,OAAO,I;MAC1B,IAAI,CAAC,UAAL,C;QAAiB,OAAO,K;MAExB,gBA  
AqB,cAAL,SAAK,C;MACrB,iBAAuB,cAAN,KAAM,C;MAEHb,yBAAa,U;MAAb,U;QAA2B,OfRMyB,oBEqMz  
B,SFrMyB,CAAY,cAfrB,YAAY,CAAZ,CEoNhB,KFrMyB,oBEqMI,UFrMJ,CAAY,cAfrB,YAAY,CAAZ,C;;MEo  
NID,W;K;IAGJ,gC;MAGyC,QAAQ,cAAA,sCAAk,cAAL,EAAoB,sCAAk,cAAzB,CAAR,6B;K;IuL3OzC,6C;MA

e6B,4B;QAAA,eAAuB,G;MACHD,wCAASB,EAAtB,EAA0B,YAA1B,C;K;IAEJ,mE;MAKwC,yB;QAAA,YAAoB ,E;MAAI,4B;QAAA,eAAuB,G;MrMEnF,IAAI,CmBwR+C,CAAC,QkLzR5C,YILyR4C,CnBxRpD,C;QACI,cqMFi C,wC;QrMGjC,MAAM,gCAAyB,OAAQ,WAAjC,C;;MqMFV,cAAy,gB;MAEC,yBAAS,mBAAS,YAAA,SAAU, OAAV,EAAMb,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAAkB,SAAIB,C;MA2E9D,gBAAGB,iBA3ET,OA2 ES,C;M/Lg7CT,kBAAoB,gB;MAwSd,gB;MADb,YAAY,C;MACC,O+LnyDN,O/LmyDM,W;kBAAb,OAAa,cAAb ,C;QAAa,sB;QA9RsB,U;QAAA,cA8RT,oBAAMb,cAAnB,EAAMb,sBAAnB,U;Q+LntDIB,kB;;YAHA,CAAC,Y AAS,CAAT,IAAc,qBAAf,KAA4C,Q/LstDG,I+LttDH,C;UAC5C,a;;UAEA,4B;UA/E+B,uB;;YnLgHzB,kC;YAAA, wBZkrDyC,IYlrDzC,C;YAAA,qB;YAAA,oB;YAAA,oB;YAAAd,gE;cACI,ImLjHkD,CAAI,aAAH,UnLiHrC,YZirD qC,IYjrDrC,YAAK,OAAL,EmLjHqC,CAAG,CnLiHtD,C;gBACI,sBAAO,O;gBAAP,wB;;;YAGR,sBAAO,E;;;Um LrHH,iD;UAGI,gCAA2B,EAA3B,C;YAHJ,2BAGqC,I;iBACjC,IAAK,a/L8xD0C,I+L9xD1C,gBAAYB,uBAAZB,C AAL,C;YAJJ,2B/LkyDmD,IO9kDsB,WwLhNI,0BAAuC,mBAAvC,IxLgNJ,C;;YwLpNzE,2BAKY,I;UA0ER,iEl MND,yBkMMC,4B/LmtD+C,I;;QA9RpB,8B;UAA6C,6B;;;M+LrgDhF,OAKFK,S/Lo7CE,W+Lp7CF,EAAO,mBA Ac,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAxET,+B;MAGByC,gCAAc,EAAd,C;K;IAEzC,6C;MAGgC,yB;Q AAA,YAAoB,E;MAM3C,Q;MALL,cAAy,gB;M/LurBL,kBAAS,gB;MA2FA,U;MAAA,S+LhxBM,O/LgxBN,W; MAAhB,OAAgB,gBAAhB,C;QAAGB,2B;QAAM,Ia7hB6B,CAAC,Qb6hBhB,Oa7hBgB,Cb6hB9B,C;UAAwB,WA AY,WAAl,OAAJ,C;;M+L9wBrD,kB/L+wBE,W;MAMrBA,oBAAM,iBAAa,qCAAwB,EAAXB,CAAb,C;MAUEA, U;MAAA,+B;MAAb,OAAa,gBAAb,C;QAAa,wB;QACT,aAAy,uBAAC,IAAd,E;;M+L5gDhB,sBAASB,CAGjB,o B/L0gDE,a+L1gDF,CAHiB,mBAGF,C;MAEP,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAMb,OAAM,KAAzB, CAAT,I;MAAT,wBAAiD,kBAAkB,SAAIB,C;MAMc9D,gBAAGB,iBAnCT,OAmCS,C;M/Lg7CT,oBAAoB,gB;M AwSd,kB;MADb,YAAY,C;MACC,S+L3vDN,O/L2vDM,W;MAAb,OAAa,gBAAb,C;QAAa,0B;QA9RsB,U;QAA A,cA8RT,oBAAMb,cAAnB,EAAMb,sBAAnB,U;Q+LntDIB,kB;Q/Lq7C2B,c+Lx7C3B,CAAC,YAAS,CAAT,IAA c,qBAAf,KAA4C,Q/LstDG,M+LttDH,C/Lw7CjB,G+Lv7C3B,I/Lu7C2B,G+Lr7C3B,oBAxCmG,Q/L2vDpD,M+L3 vDoD,kBAwCnG,YIMND,yBkMMC,4B/LmtD+C,MA9RpB,U;UAA6C,+B;;;M+L79ChF,OA0CK,S/Lo7CE,a+Lp7 CF,EAAO,mBAAC,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAjCI,8C;MAAA,qB;QAEG,IAAG,QAAH,EAAG ,CAAH,C;UAEQ,IAAA,EAAG,OAAH,GAAY,cAAO,OAAAnB,C;YAHZ,OAGyC,c;;YAHZC,OAIoB,E;;UAJpB,O AOY,iBAAS,E;O;K;IAfjC,0C;MAKgC,sB;QAAA,SAAiB,M;MAC7C,OAYK,eAXA,OADL,uBACK,EAAl,4BAA J,CAWA,EAAa,IAAb,C;K;IAET,gC;MAAwC,uB;;QnLkDtB,gC;QAAA,gC;QAAA,mB;QAAA,kB;QAAA,kB;QA Ad,0D;UACI,ImLnD+C,CAAI,aAAH,UnLmDlC,iCAAK,KAAL,EmLnDkC,CAAG,CnLmDnD,C;YACI,sBAAO, K;YAAP,wB;;;QAGR,sBAAO,E;;;Mf5CA,4B;MkMX6B,OAA8C,OAAM,EAAV,GAAC,gBAAd,GAA0B,E;K;IA GpF,wC;MAAkB,W;K;IAC9B,oD;MAAA,uB;QAakB,wBAAS,I;O;K;IAFvC,mC;MACI,IAAA,MILgMgD,YAAU ,CkLhM1D,C;QAD4C,OACxB,wB;;QADwB,OAEPc,kC;K;mBAGZ,yB;M/L86CA,+D;MAwSA,wE;M+LttDA,sF; QAKI,gBAAGB,2B;Q/Lg7CT,kBAAoB,gB;QAwSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa, sB;UA9RsB,U;UAAA,cA8RT,oBAAMb,cAAnB,EAAMb,sBAAnB,U;U+LntDIB,kB;U/Lq7C2B,c+Lx7C3B,CAA C,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q/LstDG,I+LttDH,C/Lw7CjB,G+Lv7C3B,I/Lu7C2B,G+Lr7C3B,sC/LmtD +C,I+LntD/C,alMND,yBkMMC,4B/LmtD+C,IA9RpB,U;YAA6C,6B;;;Q+Lz7ChF,OAMK,S/Lo7CE,W+Lp7CF,E AAO,mBAAC,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;O;KAbT,C;6E9EgSA,0B;MAGmE,OAAA,SAAK,gBAA O,GAAP,C;K;qFAExE,yB;MAAA,yD;MAAA,gC;QAO2B,gBAAhB,oB;QAASB,apHrU7B,W;QoHqUA,OpHpUO ,SoHoUqC,W;O;KAPhD,C;uFAUA,yB;MAAA,iE;MAAA,0C;QAQmC,gBAAXB,mBAAC,QAAd,C;QAA8B,apHh VrC,W;QoHgVA,OpH/uo,SoH+U6C,W;O;KARxD,C;IAWA,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAa,WA AA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;IAGX,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAa, WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;6EAGX,yB;MAAA,2D;MAAA,8C;QAI+F,MAAM, 8B;O;KAJrG,C;qFAMA,qB;MAG8D,gCAAO,EAAP,C;K;qFAE9D,4B;MAGkF,OAAA,yBAAO,KAAP,CALpB,g BAAO,EAAP,C;K;qFAO9D,4B;MAG4E,OAAA,yBAAO,KAAP,CAVd,gBAAO,EAAP,C;K;qFAY9D,4B;MAGy E,OAAA,yBAAO,KAAP,CAfX,gBAAO,EAAP,C;K;qFAiB9D,4B;MAG8E,OAAA,yBAAO,KAAP,CAPhB,gBA AO,EAAP,C;K;qFAsB9D,4B;MAGyE,OAAA,yBAAO,KAAP,CAzBX,gBAAO,EAAP,C;K;qFA2B9D,4B;MAG4 E,OAAA,yBAAO,KAAP,CA9Bd,gBAAO,EAAP,C;K;I7Hrb9D,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MA QyC,IAAtB,I;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;MAA5C,UAAU,I;MACV,IA AI,MAAM,sCAAK,UAAx,IAAwB,MAAM,sCAAK,UAAvC,C;QAakD,OAAO,I;MACzD,OAAW,OAAJ,GAAl,

C;K;IAGf,kC;MAK4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MAQyC,IAAtB,I;MAAA,qBAAL,SAAK,EAAY,KAAZ,  
C;MAAL,iB;QAA2B,OAAO,I;MAA5C,UAAU,I;MACV,IAAI,MAAM,uCAAM,UAAZ,IAAyB,MAAM,uCAAM,  
UAAzC,C;QAAoD,OAAO,I;MAC3D,OAAW,QAAJ,GAAI,C;K;IAGf,gC;MAKwC,gCAAoB,EAAPB,C;K;IAExC,  
yC;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;  
MACA,c;MACA,S;MAEA,gBAAGB,qBAAK,CAAL,C;MACHB,IAAI,YAAY,EAAbB,C;QACI,IAAI,WAAU,CA  
Ad,C;UAAiB,OAAO,I;QAExB,QAAQ,C;QAER,IAAI,cAAa,EAAB,B,C;UACI,aAAa,I;UACb,QAAQ,W;eACL,IAA  
I,cAAa,EAAB,B,C;UACH,aAAa,K;UACb,QAAQ,W;UAER,OAAO,I;QAEZ,QAAQ,C;QACR,aAAa,K;QACb,QA  
AQ,W;MAIZ,uBAAB,S;MAEvB,qBAAB,qB;MACrB,aAAa,C;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QAC  
I,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAB,B,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,I  
AAI,SAAS,cAAb,C;UACI,IAAI,mBAAB,kB,BAAtB,C;YACI,iBAAB,iB,QAAQ,KAAR,I;YAEjB,IAAI,SAAS,cAAb,  
C;cACI,OAAO,I;YAGX,OAAO,I;QAI,6BAAU,KAAV,C;QAEA,IAAI,UAAS,QAAQ,KAAR,IAAT,CAAJ,C;U  
AA4B,OAAO,I;QAEnC,kBAAB,UAAV,I;MAGJ,OAAW,UAAJ,GAAgB,MAAhB,GAA4B,CAAC,MAAD,I;K;IA  
GvC,iC;MAK0C,iCAAqB,EAAR,B,C;K;IAE1C,0C;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI  
,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAGB,qBAAK,CAAL,C;MACHB,IA  
AI,YAAY,EAAbB,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAExB,QAAQ,C;QAER,IAAI,cAAa,EAAB,B  
,C;UACI,aAAa,I;UACb,gC;eACG,IAAI,cAAa,EAAB,B,C;UACH,aAAa,K;UACb,6B;UAEA,OAAO,I;QAEZ,QA  
AQ,C;QACR,aAAa,K;QACb,6B;MAIJ,2C;MAEA,qBAAB,qB;MACrB,e;MACA,aAAU,KAAV,MAAsB,MAAt  
B,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAB,B,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,  
I;QACtB,IAAI,uBAAS,cAAT,KAJ,C;UACI,IAAI,uBAAB,kB,BAAB,iB,CAAJ,C;YACI,iBAAB,iB,8BAAQ,KAAR,E;  
YAEjB,IAAI,uBAAS,cAAT,KAJ,C;cACI,OAAO,I;YAGX,OAAO,I;QAI,6CAAU,KAAV,E;QAEA,IAAI,uB  
AAS,8BAAQ,KAAR,EAAT,KAJ,C;UAA4B,OAAO,I;QAEnC,6CAAU,KAAV,E;MAGJ,OAAW,UAAJ,GAAgB  
,MAAhB,GAA6B,MAAD,a;K;IAIvC,kC;MAAyD,MAAM,0BAAsB,6BAA0B,KAA1B,MAAtB,C;K;uEyBhI/D,yB  
;MAAA,oC;MAAA,uC;QAI,iBAAB,iB,C;QACjB,eAAe,mBAAS,CAAT,I;QACf,iBAAB,iB,K;QAEjB,OAAO,cAAc,  
QAAR,B,C;UACI,YAAGB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;UAC7C,YAAY,UAAU,iCAAK,KAAL,EAAB  
,C;UAEZ,IAAI,CAAC,UAAAL,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;CAEb,0BAAC,CAAd,I;YAEJ,IAAI,C  
AAC,KAAL,C;cACI,K;CAEA,sBAAY,CAAZ,I;QAIZ,OAAO,8BAAY,UAAZ,EAAB,WAAW,CAAX,IAAxB,  
C;O;KAZBX,C;yEA4BA,yB;MAAA,8B;MA5BA,oC;MA4BA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QA5BD,iBAAB,iB,  
C;QACjB,eAAe,qBAAS,CAAT,I;QACf,iBAAB,iB,K;QAEjB,OAAO,cAAc,QAAR,B,C;UACI,YAAGB,CAAC,UAAAL,  
GAAiB,UAAjB,GAAiC,Q;UAC7C,YAsBwB,SAtBZ,CAAU,mCAAK,KAAL,EAAB,C;UAEZ,IAAI,CAAC,UAA  
L,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;CAEb,0BAAC,CAAd,I;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;CA  
EA,sBAAY,CAAZ,I;QAWZ,OAPO,gCAAY,UAAZ,EAAB,WAAW,CAAX,IAAxB,CAOGC,W;O;KAJ3C,C;IF  
AMA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIuB,UAAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SA  
AK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAB,CAAL,C;Y  
ACI,OAAO,8BAAY,KAAZ,EAAB,mB,gBAAnB,C;QAEf,OAAO,E;O;KARX,C;mFAWA,yB;MAAA,8B;MAXA,m  
D;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,oB;UAXJ,kC;UAAA,qBAAL,WAAK,C;UAAAL,  
qB;UAAA,oB;UAAA,oB;UAAAd,0D;YACI,IAAI,CAUyB,SAVxB,CAAU,mCAAK,KAAL,EAAB,CAAL,C;cACI,  
mBAAO,gCAAY,KAAZ,EAAB,mB,kBAAnB,C;cAAP,qB;UAER,mBAAO,E;QAOP,OAA4C,2B;O;KAJhD,C;6E  
AMA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,C  
AAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAB,CAAL,C;YACI,OA  
AO,8BAAY,CAAZ,EAAB,e,QAAQ,CAAR,IAAf,C;QAEf,OAAO,E;O;KARX,C;+EAWA,yB;MAAA,8B;MAXA,m  
D;MAAA,+C;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,kB;UAXT,U;UAAA,SAAa,SAAR,Y  
AAL,WAAK,CAAQ,CAAb,W;UAAAd,OAAc,gBAAd,C;YAAc,yB;YACV,IAAI,CAUyB,SAVtB,CAAU,mCAAK,  
KAAL,EAAB,CAAL,C;cACI,iBAAO,gCAAY,CAAZ,EAAB,e,QAAQ,CAAR,IAAf,C;cAAP,mB;UAER,iBAAO,E;  
QAOP,OAA0C,yB;O;KAJ9C,C;IAMA,kC;MAhEI,iBAAB,iB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAB,iB,K;  
MAEjB,OAAO,cAAc,QAAR,B,C;QACI,YAAGB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YA6DgE,4BA  
7D1C,iCAAK,KAAL,EA6D0C,E;QA3DhE,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;Y  
AEb,0BAAC,CAAd,I;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;YAEA,sBAAY,CAAZ,I;MAkDiD,OA9CtD,8BAA  
Y,UAAZ,EAAB,WAAW,CAAX,IAAxB,C;K;IAGDX,kC;MAzCK,Q;MAAsB,kBAAtB,2D;MA5BD,iBAAB,iB,C;

MACjB,eAAe,qBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAL,  
GAAiB,UAAjB,GAAiC,Q;QAC7C,YAkEoD,4BAIE9B,mCAAK,KAAL,EAkE8B,E;QAhEpD,IAAI,CAAC,UAAL  
,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YA  
EA,sBAAY,CAAZ,I;;MAuDqC,OAnD1C,gCAAY,UAAZ,EAawB,WAAW,CAAX,IAAxB,CAOGC,W;K;IA8C3  
C,uC;MAGsE,oB;;QA3C/C,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CA0CsE,4BA  
1C3D,iCAAK,KAAL,EA0C2D,EA1C1E,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;QAE  
R,mBAAO,E;;MAuC2D,uB;K;IAEtE,uC;MAICK,Q;MAAsB,kBAAtB,2D;MAAsB,oB;;QAXJ,kC;QAAA,wBAA  
L,WAAK,C;QAAL,qB;QAAA,oB;QAAA,oB;QAAd,0D;UACI,IAAI,CA+C0D,4BA/C/C,mCAAK,KAAL,EA+C+  
C,EA/C9D,C;YACI,mBAAO,gCAAY,KAAZ,EAAmB,kBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MA4C+C,OAr  
CV,2B;K;IAuChD,qC;MAGoE,kB;;QApCID,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAc,C;U  
AAc,uB;UACV,IAAI,CAmCkE,4BAnCvD,iCAAK,KAAL,EAuCuD,EAnCtE,C;YACI,iBAAO,8BAAY,CAAZ,E  
AAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;MAGCyD,qB;K;IAEpE,qC;MA3BK,Q;MAAsB,kBA  
AtB,2D;MAAsB,kB;;QAXT,U;QAAA,SAAa,WAAR,eAAL,WAAK,CAAQ,CAAb,W;QAAd,OAAc,gBAAd,C;UA  
Ac,yB;UACV,IAAI,CAwCsD,4BAxC3C,mCAAK,KAAL,EAwC2C,EAxC1D,C;YACI,iBAAO,gCAAY,CAAZ,E  
AAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;MAqC6C,OA9BV,yB;K;IAGC9C,2B;MA9FI,iBAAiB,  
C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UA  
L,GAAiB,UAAjB,GAAiC,Q;QAC7C,mCAAsB,iCAAK,KAAL,EAAtB,E;QAEA,IAAI,CAAC,UAAL,C;UACI,IA  
AI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY  
,CAAZ,I;;MAGf+B,OA5EpC,8BAAY,UAAZ,EAawB,WAAW,CAAX,IAAxB,C;K;yEA8EX,yB;MAAA,8B;MA  
AA,qC;MAAA,4B;QAI2C,Q;QAAD,OAAuB,KAAtB,2DAAsB,CAAO,W;O;KAJxE,C;IAMA,gC;MAGoD,oB;;Q  
A1E7B,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,wBAAW,iCAAK,KAAL,EAAX,E  
AAJ,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MASyC,uB;K;mFA  
EpD,yB;MAAA,8B;MAAA,+C;MAAA,4B;QAIgD,Q;QAAD,OAAuB,UAAtB,2DAAsB,CAAY,W;O;KAJIF,C;IA  
MA,8B;MAGkD,kB;;QApEhC,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAc,C;UAAc,uB;UAC  
V,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAJ,C;YACI,iBAAO,8BAAY,CAAZ,EAae,QAAQ,CAAR,IAAf,C;YA  
AP,mB;;QAER,iBAAO,E;;MAGEuC,qB;K;+EAEID,yB;MAAA,8B;MAAA,2C;MAAA,4B;QAI8C,Q;QAAD,OA  
AuB,QAAtB,2DAAsB,CAAU,W;O;KAJ9E,C;IAMA,8C;MAU8C,uB;QAAA,UAAgB,E;MAO5C,Q;MAND,IAAI,  
SAAS,CAAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;Q  
ACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAae,SAAK,OAApB,C;MAEHb,SAAS,mBAAc,MAAd,C;MACK,gB  
AAS,SAAK,OAAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAp,C;MACP,EAAG,gBAAO,SAAP,C;M  
ACH,OAAO,E;K;IAGX,gD;MASwC,uB;QAAA,UAAgB,E;MACnD,Q;MAAD,OAAuB,SAAtB,6DAAsB,EAAS,  
MAAT,EAiB,OAAjB,CAA0B,W;K;IAErD,4C;MAU4C,uB;QAAA,UAAgB,E;MAQ1C,Q;MAPd,IAAI,SAAS,C  
AAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OA  
AY,mBAAL,SAAK,EAAY,CAAZ,EAae,SAAK,OAApB,C;MAEHb,SAAS,mBAAc,MAAd,C;MACT,EAAG,gBA  
AO,SAAP,C;MACW,gBAAS,SAAK,OAAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAp,C;MACP,O  
AAO,E;K;IAGX,8C;MASSC,uB;QAAA,UAAgB,E;MACjD,Q;MAAD,OAAuB,OAAtB,6DAAsB,EAEO,MAAP,E  
AAe,OAAf,CAAwB,W;K;2FAEnD,qB;MAWI,OAAO,qBAAgB,SAAK,OAAL,KAAe,C;K;+EAG1C,qB;MAMoD,  
4BAAU,C;K;sFAE9D,qB;MAMuD,0BAAS,C;K;mFAMhE,yB;MAAA,2C;MAAA,4B;QAMuD,QAAC,kB;O;KA  
NxDC,yFAQA,yB;MAAA,2C;MAAA,4B;QAWI,OAAO,qBAAqB,QAAL,SAAK,C;O;KAXhC,C;IAiB4D,+C;MA  
AA,kC;MAAS,uB;MACjE,eAAoB,C;K;gDAEpB,Y;MAA2C,gB;MAAA,iE;MAAJ,4C;K;+CAEvC,Y;MAAyC,sB  
AAQ,yB;K;;IARrD,+B;MAG4D,4C;K;+EAQ5D,qB;MAE8C,uCAAQ,E;K;+EAEtD,mC;MASI,OA5DgD,qBAAU,  
CA4D1D,GAAe,cAAf,GAAmC,S;K;6EAEvC,yB;MAAA,2C;MAAA,0C;QASI,OAAI,kBAAJ,GAAe,cAAf,GAAm  
C,S;O;KATvC,C;IAeI,mC;MAAQ,uBAAG,mBAAS,CAAT,IAAH,C;K;IAMR,qC;MAAQ,OAAA,SAAK,OAAL,G  
AAc,CAAd,I;K;IAEZ,8C;MAIuB,Q;MAAA,0BAAS,CAAT,I;MAAnB,OAAgB,CAAT,8BACgB,gBAAZ,qBAAK,  
KAAL,CAAY,CADhB,IAEoB,eAAhB,qBAAK,QAAQ,CAAR,IAAL,CAAgB,C;K;IAG/B,uC;MAGuD,ON3IyC,o  
BM2I/B,KAAM,MN3IyB,EM2IIB,KAAM,aAAN,GAAqB,CAArB,IN3IkB,C;K;IM6IhG,yC;MAGqE,qCAAY,KA  
AM,MAAIB,EAyB,KAAM,aAAN,GAAqB,CAArB,IAAzB,C;K;uFAErE,iC;MAS2E,2BAAY,KAAZ,EAAmB,G  
AAnB,C;K;mFAE3E,2C;MAO0D,wB;QAAA,WAAGB,gB;MAAkB,OAAA,8BAAY,UAAZ,EAawB,QAAxB,CA

AkC,W;K;IAE9H,uC;MAG6D,OAAA,8BAAY,KAAM,MAAIB,EAAYB,KAAM,aAAN,GAAqB,CAArB,IAAzB,C  
AAiD,W;K;IAE9G,sE;MAImD,qC;QAAA,wBAAGC,S;MAC/E,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,E  
AApB,GAAwB,qBAAXB,GNjL4F,oBMiL/B,CNjL+B,EMiL5B,KNjL4B,C;K;IMoLhG,wE;MAIqD,qC;QAAA,wB  
AAgC,S;MACjF,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAXB,GN1L4F,oBM0L/B,  
CN1L+B,EM0L5B,KN1L4B,C;K;IM6LhG,qE;MAIkD,qC;QAAA,wBAAGC,S;MAC9E,YAAY,sBAAQ,SAAR,C;  
MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAXB,GNnM4F,oBMmM/B,QAAQ,CAAR,INnM+B,EMmMpB,gBN  
nMoB,C;K;IMsMhG,uE;MAIoD,qC;QAAA,wBAAGC,S;MACHf,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,  
EAAPB,GAAwB,qBAAXB,GN5M4F,oBM4M/B,QAAQ,SAAU,OAAIB,IN5M+B,EM4ML,gBN5MK,C;K;IM+Mh  
G,0E;MAIuD,qC;QAAA,wBAAGC,S;MACnF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,  
qBAAXB,GNrN4F,oBMqN/B,CNrN+B,EMqN5B,KNrN4B,C;K;IMwNhG,4E;MAIyD,qC;QAAA,wBAAGC,S;MA  
CrF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAXB,GN9N4F,oBM8N/B,CN9N+B,E  
M8N5B,KN9N4B,C;K;IMiOhG,yE;MAIsD,qC;QAAA,wBAAGC,S;MACIF,YAAY,0BAAY,SAAZ,C;MACZ,OA  
AW,UAAS,EAAPB,GAAwB,qBAAXB,GNvO4F,oBMuO/B,QAAQ,CAAR,INvO+B,EMuOpB,gBNvOoB,C;K;IM  
0OhG,2E;MAIwD,qC;QAAA,wBAAGC,S;MACpF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAA  
wB,qBAAXB,GNhP4F,oBMgP/B,QAAQ,SAAU,OAAIB,INhP+B,EMgPL,gBNhPK,C;K;IMmPhG,oE;MAOI,IAAI  
,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAKD,UAAID,OAA1B,C;MACV,SAAS,sB;MACT,  
EAAG,qBAAY,SAAZ,EAakB,CAAIB,EAaqB,UAArB,C;MACH,EAAG,gBAAO,WAAp,C;MACH,EAAG,qBA  
AY,SAAZ,EAakB,QAAlB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;yFAGX,yB;MAAA,8B;MAAA,qD;MAAA,+  
D;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,UAAb,EAAYB,QAazB,EAAMc,WAAnc,CAAGD,W;O;KA  
P3E,C;IASA,uD;MAOI,+BAAa,KAAM,MAAnB,EAA0B,KAAM,aAAN,GAAqB,CAArB,IAA1B,EAakD,WAAI  
D,C;K;yFAEJ,yB;MAAA,8B;MAAA,qD;MAAA,gD;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,KAAb,EA  
AoB,WAApB,CAAIc,W;O;KAP5D,C;IASA,sD;MASI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,Q  
AAb,oCAAKD,UAAID,OAA1B,C;MAEV,IAAI,aAAY,UAAhB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,E  
AAe,gBAAf,C;MAEhB,SAAS,mBAAc,oBAAU,QAAY,GAAqB,UAArB,KAAc,C;MACT,EAAG,qBAAY,SAAZ,  
EAakB,CAAIB,EAaqB,UAArB,C;MACH,EAAG,qBAAY,SAAZ,EAakB,QAAlB,EAA4B,gBAA5B,C;MACH,O  
AAO,E;K;uFAGX,yB;MAAA,8B;MAAA,mD;MAAA,kD;QASK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,UA  
AZ,EAawB,QAaxB,CAakC,W;O;KAT7D,C;IAWA,yC;MAKqE,8BAAY,KAAM,MAAIB,EAAYB,KAAM,aAA  
N,GAAqB,CAArB,IAAzB,C;K;uFAErE,yB;MAAA,8B;MAAA,mD;MAAA,mC;QAOK,Q;QAAD,OAAuB,YAAt  
B,2DAAsB,EAAY,KAAZ,CAAmB,W;O;KAP9C,C;IASA,yC;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,OAA  
O,8BAAY,MAAO,OAAAnB,EAA2B,gBAA3B,C;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MA  
KI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,ONiWyE,oBMkWXD,MAAO,ONiWiD,C;;MMoW7E,OAAO,S;K;IAG  
X,yC;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,OAAO,8BAAY,CAAZ,EAAe,mBAAS,MAAO,OAAhB,IAAf,C  
;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,ONrX  
wF,oBMqXvE,CNrXuE,EMqXpE,mBAAS,MAAO,OAAhB,INrXoE,C;;MMuX5F,OAAO,S;K;IAGX,sD;MAMI,I  
AAK,qBAAU,MAAO,OAAP,GAAgB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAMe,s  
BAAS,MAAT,CAAvE,C;QACI,OAAO,8BAAY,MAAO,OAAAnB,EAA2B,mBAAS,MAAO,OAAhB,IAA3B,C;;M  
AEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,wD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAgB,MAAO  
,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAMe,sBAAS,MAAT,CAAvE,C;QACI,ON7YwF,oB  
M6YvE,MAAO,ON7YgE,EM6YxD,mBAAS,MAAO,OAAhB,IN7YwD,C;;MM+Y5F,OAAO,S;K;IAGX,mD;MA  
Kmf,oCAAKB,SAAlB,EAA6B,SAA7B,C;K;IAEnF,mD;MAKuE,sCAAKB,SAAlB,EAA6B,SAA7B,C;K;IAEvE,iF  
;MAIsE,qC;QAAA,wBAAGC,S;MACIG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA  
,OAAiB,qB;;QA5JvB,U;QA4JM,OA5JgB,aAAtB,+DAAsB,EA4JyC,CA5JzC,EA4J4C,KA5J5C,EA4JmD,WA5Jn  
D,CAAGD,W;;MA4JvE,W;K;IAGJ,mF;MAIwE,qC;QAAA,wBAAGC,S;MACpG,YAAY,sBAAQ,SAAR,C;MACL,  
Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QArKvB,U;QAqKM,OArKgB,aAAtB,+DAAsB,EAqKyC,CA  
rKzC,EAqK4C,KArK5C,EAqKmD,WArKnD,CAAGD,W;;MAqKvE,W;K;IAGJ,gF;MAIqE,qC;QAAA,wBAAGC,S  
;MACjG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,Q  
AAQ,CAAR,I;QAAb,eAAwB,gB;QA9K1E,U;QA8KM,OA9KgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,  
EA8K4D,WA9K5D,CAAGD,W;;MA8KvE,W;K;IAGJ,kF;MAIuE,qC;QAAA,wBAAGC,S;MACnG,YAAY,sBAAQ

,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;Q  
AAb,eAAuC,gB;QAvLzF,U;QAUlM,OAuLgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAuL2E,WAvL3  
E,CAAgD,W;;MAuLvE,W;K;IAGJ,oF;MAI2E,qC;QAAA,wBAAGC,S;MACvG,YAAY,0BAAY,SAAZ,C;MACL,  
Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;Q  
AhMzF,U;QAgMM,OAHMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAgm2E,WAhM3E,CAAgD,W;;  
MAGmV,E,W;K;IAGJ,sF;MAIyE,qC;QAAA,wBAAGC,S;MACrG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAA  
I,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QAzM1E,U;QAYMM,O  
AzMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAyM4D,WazM5D,CAAgD,W;;MAYmV,E,W;K;IAGJ,q  
F;MAI0E,qC;QAAA,wBAAGC,S;MACtG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAA  
A,OAAiB,qB;;QAINvB,U;QAKNM,OAlNgB,aAAtB,+DAAsB,EAkNyC,CAINzC,EAkN4C,KAIN5C,EAkNmD,W  
AlNnD,CAAgD,W;;MAkNvE,W;K;IAGJ,uF;MAI4E,qC;QAAA,wBAAGC,S;MACxG,YAAY,0BAAY,SAAZ,C;M  
ACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QA3NvB,U;QA2NM,OA3NgB,aAAtB,+DAAsB,EA2N  
yC,CA3NzC,EA2N4C,KA3N5C,EA2NmD,WA3NnD,CAAgD,W;;MA2NvE,W;K;+EAOJ,yC;MAQoF,OAAA,KA  
AM,iBAAQ,SAAR,EAAC,WAAc,C;K;+EAE1F,uC;MAOI,OAAA,KAAM,iBAAQ,SAAR,EAAC,SAAd,C;K;yFAE  
V,yC;MAMyF,OAAA,KAAM,sBAAa,SAAb,EAAMb,WAAAnB,C;K;+FAE/F,yB;MAAA,oC;MAAA,gC;MAAA,u  
C;QAeW,Q;QAAA,IApe4C,mBAAS,CAoerD,C;uBAAkB,oBAAU,iCAAK,CAAL,EAav,E;UAAA,YNzhBoD,oB  
MyhBrB,CNzhBqB,C;UMyhBtE,OLrjBwD,2BAAL,GAakB,K;;UKqjBrE,OAAYD,S;QAaHE,W;O;KafJ,C;iGakB  
A,yB;MAAA,oC;MAAA,uC;QAeI,OAftmD,mBAAS,CAsf5D,GAAYB,UAAU,iCAAK,CAAL,EAav,CAAmB,W  
AAAnB,GN3iBoD,oBM2iBV,CN3iBU,CM2iB7E,GAA2E,S;O;Kaf/E,C;+EAmBA,4B;MAIsE,OAAA,KAAM,iBA  
AQ,SAAR,C;K;IAE5E,OF;MAKI,IAAK,cAAc,CAaf,IAAsB,aAAa,CAAnC,IAA0C,cAAa,SAAK,OAAL,GAAC,M  
AAd,IAAb,CAA1C,IAAiF,eAAc,KAAM,OAAN,GAAe,MAAf,IAAd,CAArF,C;QACI,OAAO,K;;MAGX,iBAAC,C  
AAd,UAAsB,MAAtB,U;QACI,IAAI,CAA0B,SAazB,qBAAK,aAAa,KAAb,IAAL,CAAYB,EAAO,iBAAM,cAAc,  
KAAd,IAAN,CAAP,EAAMC,UAAAnC,CAA9B,C;UACI,OAAO,K;;MAEf,OAAO,I;K;IAGX,mD;MAG+C,0B;QA  
AA,aAAsB,K;MACjE,OAAA,SAAK,OAAL,GAAC,CAAd,IAA2B,SAAR,qBAAK,CAAL,CAAQ,EAAO,IAAP,E  
AAa,UAAb,C;K;IAE/B,iD;MAG6C,0B;QAAA,aAAsB,K;MAC/D,OAAA,SAAK,OAAL,GAAC,CAAd,IAAMC,S  
AAhB,qBAAK,2BAAL,CAAqB,EAAO,IAAP,EAAa,UAAb,C;K;IAEvC,qD;MAGyD,0B;QAAA,aAAsB,K;MAC3  
E,IAAI,CAAC,UAAD,IAAe,6BAaf,IAAiC,0BAArC,C;QACI,OAAY,WAAL,SAAK,EAaw,MAAX,C;;QAEZ,O  
AAO,6BAakB,CAAIB,EAAqB,MAArB,EAA6B,CAA7B,EAAgC,MAAO,OAAvC,EAA+C,UAA/C,C;K;IAGf,iE;  
MAG0E,0B;QAAA,aAAsB,K;MAC5F,IAAI,CAAC,UAAD,IAAe,6BAaf,IAAiC,0BAArC,C;QACI,OAAY,aAAL,  
SAAK,EAaw,MAAX,EAAMb,UAAAnB,C;;QAEZ,OAAO,6BAakB,UAAIB,EAA8B,MAA9B,EAAc,CAAAtC,EA  
AyC,MAAO,OAahD,EAawD,UAAxD,C;K;IAGf,mD;MAGuD,0B;QAAA,aAAsB,K;MACzE,IAAI,CAAC,UAA  
D,IAAe,6BAaf,IAAiC,0BAArC,C;QACI,OAAY,SAAL,SAAK,EAAS,MAAT,C;;QAEZ,OAAO,6BAakB,mBAA  
S,MAAO,OAahB,IAAIB,EAA0C,MAA1C,EAakD,CAAID,EAAqD,MAAO,OAA5D,EAAoE,UAApE,C;K;IAMf,  
wD;MAQ8D,0B;QAAA,aAAsB,K;MACHf,qBfjnBO,MAAO,KeinBa,SAAK,OfjnBIB,EeinB0B,KAAM,OfjnBhC,  
C;MemnBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA8B,SAAR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,C  
AAP,EAA8B,UAA9B,CAArC,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,IAAI,CAAJ,IAAnB,CAAL,IAA  
wC,mBAAN,KAAM,EAAMb,IAAI,CAAJ,IAAnB,CAA5C,C;QACI,a;;MAEJ,OAAO,8BAAY,CAAZ,EAAe,CAAf  
,CAakB,W;K;IAG7B,wD;MAQ8D,0B;QAAA,aAAsB,K;MACHf,iBAaiB,SAAK,O;MACTb,kBAakB,KAAM,O;  
MACxB,qBfxoBO,MAAO,KewoBa,UfxoBb,EewoByB,WfxoBzB,C;Me0oBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ  
,IAA+C,SAAZB,qBAAK,aAAa,CAAb,GAAiB,CAAjB,IAAL,CAAYB,EAAO,iBAAM,cAAc,CAAd,GAakB,CAAI  
B,IAAN,CAAP,EAAGD,UAAhD,CAAtD,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,aAAa,CAAb,GAAiB,  
CAAjB,IAAnB,CAAL,IAAqD,mBAAN,KAAM,EAAMb,cAAc,CAAd,GAakB,CAAIB,IAAnB,CAAzD,C;QACI,a  
;;MAEJ,OAAO,8BAAY,aAAa,CAAb,IAAZ,EAA4B,UAA5B,CAAwC,W;K;IAMnD,8D;MAQqD,0B;QAAA,aAA  
kB,C;MAAG,0B;QAAA,aAAsB,K;MAMnE,UAAkB,M;MAL3C,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAC  
,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONjtBwF,kB2G3ME,oBrG45BrE,IqG55BqE  
,C3G2MF,EMitB7D,UNjtB6D,C;;MMotBnE,uBAAX,UAAW,EAAC,CAAd,C;MAAkB,oC;kBAA3C,gD;QACI,kB  
AAkB,qBAAL,KAJ,C;QACR,c;;UIC4mXE,U;UAAhB,4BkC5mXQ,KIC4mXR,kB;YAAgB,cAAhB,UkC5mXQ,K  
IC4mXR,S;YAAsB,IkC5mXC,SAAH,UIC4mXgB,oBkC5mXhB,CAAG,0BIC4mXD,C;cAAwB,aAAO,I;cAAP,e;;;



UAC9C,aAAO,K;;;QkC7mXH,e;UACI,OAAO,K;;MAEf,OAAO,E;K;IAGX,kE;MASyD,0B;QAAA,aAAkB,2B;M  
AAW,0B;QAAA,aAAsB,K;MACxG,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAC,CAA7B,IAAkC,6BAAc,C  
;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONruB4F,sB2G3MM,oBrGg7BzE,IqGh7ByE,C3G2MN,EMquB7D,UNr  
uB6D,C;;kBMyuBhG,iBAAYB,eAAX,UAAW,EAAa,2BAAb,CAAZB,WAAwD,CAAXD,U;QACI,kBAAkB,qBAA  
I,KAAJ,C;QACR,c;;UICoIXE,Q;UAAhB,wBkCplXQ,KICoIXR,gB;YAAgB,cAAhB,UkCplXQ,KICoIXR,O;YAA  
B,IkCplXC,SAAH,UICoIXgB,oBkCplXhB,CAAG,0BICoIXD,C;cAAwB,aAAO,I;cAAP,e;;;UAC9C,aAAO,K;;;Qk  
CrIXH,e;UACI,OAAO,K;;MAGf,OAAO,E;K;IAIX,8E;MAA2G,oB;QAAA,OAAgB,K;MAOrG,UAKA,M;MAXIB  
,cAAkB,CAAC,IAAL,GACV,aAAW,gBAAX,UAAW,EAAc,CAAd,CAAX,EAAc,eAAT,QAAS,EAAa,gBAAb,  
CAAtC,CADU,GAGV,SAAW,eAAX,UAAW,EAAa,2BAAb,CAAX,EAAmD,gBAAT,QAAS,EAAc,CAAd,CAAn  
D,C;MAEJ,IAAI,iCAAkB,yBAAtB,C;QACkB,yB;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAU,cAAN,KAAM,  
EAAc,CAAd,EAAiB,SAAjB,EAAuB,KAAvB,EAA8B,KAAM,OAAPC,EAA4C,UAA5C,CAAV,C;YACI,OAAO,  
K;;;QAGD,2B;QAAd,OAAc,gBAAd,C;UAAc,2B;UACV,IAAU,kBAAN,KAAM,EAAkB,CAAlB,EAAqB,SAArB,  
EAA2B,OAA3B,EAAkC,KAAM,OAAXC,EAAgD,UAAhD,CAAV,C;YACI,OAAO,O;;;MAGnB,OAAO,E;K;IAG  
X,qE;MAUsB,UAMA,M;MAfIB,IAAI,CAAC,UAAD,IAAe,OAAQ,KAAR,KAAGB,CAAnC,C;QACI,aAAqB,UA  
AR,OAAQ,C;QACrB,YAAgB,CAAC,IAAL,GAAW,sBAAQ,MAAR,EAAgB,UAAhB,CAAX,GAA4C,0BAA Y,M  
AAZ,EAAoB,UAApB,C;QACxD,OAAW,QAAQ,CAAZ,GAAe,IAAf,GAAYB,UAAAS,MAAT,C;;MAGpC,cAAkB,  
CAAC,IAAL,GAAW,aAAW,gBAAX,UAAW,EAAc,CAAd,CAAX,EAA6B,gBAA7B,CAAX,GAAoD,SAAW,eA  
AX,UAAW,EAAa,2BAAb,CAAX,EAA0C,CAA1C,C;MAEIE,IAAI,6BAAJ,C;QACkB,yB;oBAAd,OAAc,cAAd,C  
;UAAc,yB;UACmB,sB;;Yb3sBrB,U;YAAA,Sa2sBa,Ob3sBb,W;YAAhB,OAAgB,gBAAhB,C;cAAgB,2B;cAAM,I  
a2sBgC,cb3sBIB,Oa2sBkB,EAAc,CAAd,sBb3sBIB,Oa2sBmD,OAAjC,ab3sBhC,C;gBAAwB,qBAAO,O;gBAAP,u  
B;;;YAC9C,qBAAO,I;;;Ua0sBC,uC;UACA,IAAI,sBAAJ,C;YACI,OAAO,YAAS,cAAT,C;;;QAGD,2B;oBAAd,O  
AAc,gBAAd,C;UAAc,2B;UACmB,wB;;YbjtBrB,U;YAAA,SaitBa,ObjtBb,W;YAAhB,OAAgB,gBAAhB,C;cAAG  
B,6B;cAAM,IaitBgC,kBbjtBIB,SaitBkB,EAAkB,CAAlB,sBbjtBIB,SaitBuD,OAARc,abjtBhC,C;gBAAwB,uBAAO,  
S;gBAAP,uB;;;YAC9C,uBAAO,I;;;UagtBC,2C;UACA,IAAI,wBAAJ,C;YACI,OAAO,YAAS,gBAAT,C;;;MAInB,  
OAAO,I;K;IAGX,iE;MAY+D,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACtG,4BAAU,OA AV,EAAm  
B,UAAAnB,EAA+B,UAA/B,EAAkD,KAAID,C;K;IAEJ,mE;MAYmE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,a  
AAsB,K;MACIH,4BAAU,OA AV,EAAmB,UAAAnB,EAA+B,UAA/B,EAAkD,IAAID,C;K;IAEJ,kE;MAWgE,0B;Q  
AAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvG,gB;MAAA,8CAAU,OA AV,EAAmB,UAAAnB,EAA+B,UA  
A/B,EAAkD,KAAID,mDAAmE,E;K;IAEvE,sE;MAYoE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MA  
CnH,gB;MAAA,8CAAU,OA AV,EAAmB,UAAAnB,EAA+B,UAA/B,EAAkD,IAAID,mDAkE,E;K;IAKtE,6D;MA  
M4C,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACnF,OA AW,cAAc,gCAAzB,GACI,sBAAW,mBAA Y  
,IAAZ,CAAX,EAA8B,UAA9B,EAA0C,UAA1C,CADJ,GNz2B4F,kB2G3ME,oBrGujC5E,IqGvjC4E,C3G2MF,E  
M42BpE,UN52BoE,C;K;IM+2BhG,+D;MAQgD,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvF,OA  
W,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAgB,UAAhB,EAA4B,gBAA5B,EAAoC,UAApC,CADJ,GNx3B4F,k  
BM23B1E,MN33B0E,EM23BIE,UN33BkE,C;K;IM83BhG,iE;MAQgD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,  
aAAsB,K;MAC/F,OA AW,cAAc,gCAAzB,GACI,0BAAe,mBAA Y,IAAZ,CAAf,EAAkC,UAAIC,EAA8C,UAA9C,  
CADJ,GNp4BgG,sB2G3MM,oBrGklChF,IqGllCgF,C3G2MN,EMu4BpE,UNv4BoE,C;K;IM04BpG,mE;MAQoD,0  
B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnG,OA AW,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAg  
B,UAAhB,EAA4B,CAA5B,EAA+B,UAA/B,EAAkD,IAAID,CADJ,GNn5BgG,sBMs5B1E,MNt5B0E,EMs5BIE,U  
Nt5BkE,C;K;IMy5BpG,mD;MAM+D,0B;QAAA,aAAsB,K;MACjF,OA AI,yBAAJ,GACI,sBAAQ,KAAR,UAA4B,  
UAA5B,KAA2C,CAD/C,GAGI,sBAAQ,KAAR,EAAe,CAAf,EAAkB,gBAAIB,EAA0B,UAA1B,KAAyC,C;K;IAIj  
D,kD;MAMsD,0B;QAAA,aAAsB,K;MACxE,6BAAQ,IAAR,UAA2B,UAA3B,KAA0C,C;K;kFAE9C,4B;MAI0E,  
OAAA,KAAM,yBAAgB,SAAhB,C;K;IAM3C,yE;MACjC,oB;MACA,8B;MACA,oB;MACA,kC;K;IAG8C,sF;MA  
AA,gE;MAC1C,iBAAqB,E;MACrB,yBAAwC,WAA X,yCAAW,EAAS,CAAT,EAAY,oCAAM,OAAlB,C;MACxC  
,uBAA2B,sB;MAC3B,gBAA0B,I;MAC1B,eAAmB,C;K;0EAEnB,Y;MACI,IAAI,uBAAkB,CAAtB,C;QACI,iBAA  
Y,C;QACZ,gBAAW,I;;QAEX,IAAI,4CAAQ,CAAR,IAAa,uDAAa,yCAA1B,IAAmC,uBAAkB,yCAAM,OAAD,C  
;UACI,gBAAW,qCAAYB,iBAAN,yCAAM,CAAZB,C;UACX,uBAAkB,E;;UAEIB,YAAkB,iDAAN,yCAAM,EAA  
a,oBAAb,C;UACIB,IAAI,SAAS,IAAb,C;YACI,gBAAW,qCAAYB,iBAAN,yCAAM,CAAZB,C;YACX,uBAAkB,E

;;YAEIB,IAAK,QAAiB,KAAjB,aAAL,EAAY,SAAU,KAAV,a;YACZ,gBAAW,gCAAwB,KAAxB,C;YACX,yBA  
AoB,QAAQ,MAAR,I;YACpB,uBAAkB,0BAAwB,WAAU,CAAd,GAAiB,CAAJB,GAAwB,CAA5C,K;;;QAG1B,i  
BAAY,C;;K;oEAIpB,Y;MAKiB,Q;MAJb,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAJB,C;QACI  
,MAAM,6B;MACV,aAAa,mE;MAEb,gBAAW,I;MACX,iBAAY,E;MACZ,OAAO,M;K;uEAGX,Y;MACI,IAAI,m  
BAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;iDA9C5B,Y;MAA8C,+D;K;;IAgEU,0E;MAAA,0C;QhB1  
mCjD,SgB2mCH,sBAAW,kBAAX,EAAuB,YAAvB,EAakD,kBAAlD,C;QAAA,OAAwE,KAAK,CAAT,GAAY,I  
AAZ,GAAaB,OAAM,CAAN,C;O;K;IAdlG,iF;MAUkE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAaB,K;MAAO  
,qB;QAAA,QAAa,C;MAC7H,wBAAwB,KAAxB,C;MAEA,OAAO,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,K  
AA1C,EAAiD,gDAAjD,C;K;IAwBiD,gF;MAAA,0C;QAAkB,Q;QAAA,oCAAU,sBAAV,EAA0B,YAA1B,EAAq  
D,kBAArD,EAAwE,KAAxE,aAAaF,GAAG,UAAH,EAAe,WAAO,OAAtB,CAATf,O;O;K;IAIb9E,mF;MAC0E,0B  
;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAaB,K;MAAO,qB;QAAA,QAAa,C;MACrI,wBAAwB,KAAxB,C;MACA  
,qBAAGC,OAAX,UAAW,C;MAEHc,OAAO,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,sDAAj  
D,C;K;IAIX,wC;MnBlTCl,IAAI,EmBmtCl,SAAS,CnBntCb,CAAJ,C;QACI,cmBktCkB,8C;QnBjtCIB,MAAM,gCA  
AyB,OAAQ,WAAjC,C;;K;ImBkuCgE,sD;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IAZhF,mE;MAWmE,0B;QA  
AA,aAAaB,K;MAAO,qB;QAAA,QAAa,C;MACzG,OAAsE,OAAtE,+BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,  
KAA/D,CAAsE,EAAI,iCAAJ,C;K;IAE1E,yD;MAWyD,0B;QAAA,aAAaB,K;MAAO,qB;QAAA,QAAa,C;MAC/F,  
IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,gBAAGB,WAAW,CAAX,C;QACHb,IAAI,EAAC,SAh/BuC,YAA  
U,CAg/BID,CAAJ,C;UACI,OAAO,mBAAM,SAAN,EAAiB,UAAjB,EAA6B,KAA7B,C;;;MAI2E,kBAAb,cAAtE,  
+BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;MbgPtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb  
,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WaxTgF,uBbwTIE,IaxTkE,CbwThF,C;  
;MaxThB,ObyTO,W;K;Ia9SmE,wD;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IARhF,qE;MAOiE,0B;QAAA,aAA  
sB,K;MAAO,qB;QAAA,QAAa,C;MACvG,OAAsE,OAAtE,6BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,C  
AAsE,EAAI,mCAAJ,C;K;IAE1E,2D;MAOuD,0B;QAAA,aAAaB,K;MAAO,qB;QAAA,QAAa,C;MAC7F,IAAI,U  
AAW,OAAX,KAAmB,CAAvB,C;QACI,OAAO,mBAAoB,oBAAd,WAAW,CAAX,CAAc,CAApB,EAAGC,UAAh  
C,EAA4C,KAA5C,C;;MAG+E,kBAAb,cAAtE,6BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;Mb  
uNtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT  
,WAAY,Wa/RgF,uBb+RIE,Ia/RkE,Cb+RhF,C;;Ma/RhB,ObgSO,W;K;Ia7RX,0D;MASI,wBAAwB,KAAxB,C;MA  
EA,oBAAoB,C;MACpB,gBAAGB,sBAAQ,SAAR,EAAMb,aAAnB,EAakC,UAAIC,C;MACHb,IAAI,cAAa,EAAb  
,IAAmB,UAAaB,CAAhC,C;QACI,OAAO,OAAO,SAAK,WAAZ,C;;MAGX,gBAAGB,QAAQ,C;MACxB,aAAa,iB  
AAaB,SAAJ,GAAqB,eAAN,KAAmB,EAAa,EAAb,CAArB,GAA2C,EAA7D,C;;QAET,MAAO,WA36B6E,8BA26  
B/D,aA36B+D,EA26BhD,SA36BgD,CAAKC,WA26B/G,C;QACP,gBAAGB,YAAY,SAAU,OAAtB,I;QAEhB,IAA  
I,aAAa,MAAO,KAAP,MAAE,QAAQ,CAAR,IAAf,CAAJB,C;UAA2C,K;QAC3C,YAAY,sBAAQ,SAAR,EAAMb,  
aAAnB,EAakC,UAAIC,C;;MACP,sBAAa,EAAb,C;MAET,MAAO,WAl7BiF,8BAk7BnE,aAl7BmE,Eak7BpD,gB  
Al7BoD,CAAKC,Wak7BnH,C;MACP,OAAO,M;K;2EAGX,mC;MAOmD,qB;QAAA,QAAa,C;MAAmB,OAAA,  
KAAmB,eAAM,SAAN,EAAY,KAAZ,C;K;+FAEzF,mC;MAU6D,qB;QAAA,QAAa,C;MAAuB,OAAA,KAAmB,yB  
AAgB,SAAhB,EAAaB,KAAtB,C;K;IAEvG,iC;MAK2D,mCAAGB,MAAhB,EAAwB,IAAxB,EAA8B,IAA9B,E;K;  
IAE3D,0B;MAKgD,OAAe,UAAf,uBAAE,C;K;IAqB/D,uD;MAQsB,Q;MAPIB,IAAI,iCAAKB,yBAAtB,C;QACI,O  
AAy,SAAL,SAAK,EAAO,KAAP,EAA2B,IAA3B,C;;MAGhB,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,I  
AAI,qBAAGB,aAAhB,IAAiC,SAAK,OAAL,KAAe,KAAm,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB;MAAIB,aA  
AU,CAAV,gB;QACI,IAAI,CAAS,SAAR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,CAAP,EAA8B,IAA9B,C  
AAb,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,6C;MAQsB,Q;MAPIB,IAAI,iCAAKB,yBAAtB,C;QACI,OAAO,  
kBAAQ,KAAr,C;;MAGX,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAGB,aAAhB,IAAiC,SAAK,  
OAAL,KAAe,KAAm,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB;MAAIB,aAAU,CAAV,gB;QACI,IAAI,qBAAK,C  
AAL,MAAW,iBAAM,CAAN,CAAF,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,oC;MAU+C,QAAM,SAAN,C;a  
AC3C,M;UAD2C,OACjC,I;aACV,O;UAF2C,OAeHc,K;;UACH,MAAM,gCAAYB,mDAAGD,SAAZe,C;;K;IAGIB  
,0C;MAUsD,QAAM,SAAN,C;aACID,M;UADkD,OACxC,I;aACV,O;UAFkD,OAeVc,K;;UAFuC,OAG1C,I;;K;Im  
Lr8CZ,sB;MAAA,0B;MAII,aAC+B,e;MAC/B,cACgC,e;MACHc,WAC6B,e;MAC7B,YAC8B,e;MAC9B,eACiC,e;  
MACjC,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,eACiC,gB;MACjC,iB

ACmC,gB;MACnC,qBAEuC,gB;MACvC,sBAEwC,gB;MACxC,kBACoC,gB;MACpC,cACgC,gB;MACHc,iBAC  
mC,gB;MACnC,iBACmC,gB;MACnC,iBACmC,gB;MACnC,YAC8B,gB;MAC9B,aAC+B,iB;MAC/B,aAC+B,iB;  
MAC/B,uBACyC,iB;MACzC,wBAC0C,iB;MAC1C,sBACwC,iB;MACxC,uBACyC,iB;MACzC,wBAC0C,iB;MA  
C1C,sBACwC,iB;MACxC,cACgC,iB;MACHC,oBACsC,iB;MAcTc,cACgC,iB;MACHC,gBACKC,iB;MACIC,aAC  
+B,iB;MAC/B,mBACqC,iB;MACrC,YAC8B,iB;MAC9B,UAC4B,iB;MAC5B,mBACqC,iB;MACrC,gBACKC,iB;  
MACIC,mBACqC,iB;MACrC,sBACwC,iB;MAExC,sBAGwC,gB;MAExC,uBAGyC,gB;K;;IA7F7C,kC;MAAA,i  
C;QAAA,gB;;MAAA,0B;K;;;;;;2FCwEOC,Y;MAAQ,oCAAA,IAAb,C;K;IAiBpB,yC;MAAqB,kB;K;mIAC3C,Y;  
MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAl  
B,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,Y  
AAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACm  
D,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;  
mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;qIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,EAAIB,C;K;gDAEnD,Y;MAMoC,OAAA  
,UAAM,YAAY,iBAAQ,CAAR,EAAW,UAAM,YAAY,KAA7B,C;K;;6EvElH9D,yB;MAAA,iD;MAAA,4B;QAI4  
C,kBAAM,SAAN,C;O;KAJ5C,C;+EAMA,yB;MAAA,gD;MAAA,oC;QAI+D,kBAAM,SAAN,EAAy,MAAZ,C;O;  
KAJ/D,C;+EAMA,yB;MAAA,oC;MAAA,qC;QAIqE,sBAAM,SAAN,EAAy,OAAZ,C;O;KAJrE,C;IrY4B,4B;MA  
mBxB,gC;MAnB6C,0B;MAW7B,UAEA,MAFA,EAGA,M;MALZ,IgIjC8D,IhIiC9D,C;QACI,IAAI,kBAAJ,C;UA  
CQ,mB;UAAJ,IAAI,sEAA sB,SAAtB,EAAJ,C;YAAqC,MAAM,sBAAiB,YAAF,+CAAF,C;;UAEvC,qB;UAAJ,IA  
AI,0EAAuB,UAAvB,EAAJ,C;YAAuC,MAAM,sBAAiB,YAAF,gDAAF,C;UACzC,qB;UAAJ,IAAI,kEAA+B,mBA  
A/B,CAAJ,C;YAAwD,MAAM,sBAAiB,YAAF,mCAAF,C;;K;mFAZID,Y;MAAQ,kCAAA,CAAb,C;K;+FACU,Y;  
MAAQ,OAAA,eAAS,QAAT,GAAqB,C;K;qCACvE,Y;MAA0B,QADwB,eAAS,QAAT,GAAqB,CAC7C,MAAqB,  
C;K;sCAC/C,Y;MAA2B,QAFuB,eAAS,QAAT,GAAqB,CAE5C,MAAqB,C;K;yFACxB,Y;MAAQ,OAAI,kBAAJ,  
mF;K;IAAhC,8B;MAAA,kC;MACI,YAC4B,gB;MAE5B,gBACgC,iBAAiB,UAAjB,C;MACHC,4BAAsC,uC;K;mD  
AEtC,yC;MAGI,2BAAoB,KAApB,EAA2B,UAA3B,EAAuC,UAAvC,C;K;IJAM8B,yB;MAAA,6C;MAAA,iD;MA  
AA,4B;QAAQ,sD;O;KAAR,C;JAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;JAUE,yB;MA  
AA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;mJAKF,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;  
KAAR,C;mJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;  
MAAA,4B;QAAQ,uD;O;KAAR,C;mJAKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,  
yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAA  
Q,uD;O;KAAR,C;yIAKR,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MA  
AA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yI  
AKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;Q  
AAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;qIAKL,yB;MAAA,6C;  
MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;qIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C  
;qIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;mIAKJ,yB;MAAA,6C;MAAA,iD;MAAA,4  
B;QAAQ,+C;O;KAAR,C;mIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAUE,yB;MAA  
A,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;uDAK9B,iB;MAK+C,OAAM,WAAN,KAAM,yC;K;uDAErD  
,iB;MAKgD,OAAM,aAN,KAAM,yC;K;uDAEtD,iB;MASKD,OAAM,aAN,KAAM,yC;K;wDAGxD,iB;MAKg  
D,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM  
,aAN,KAAM,0C;K;wDAGzD,iB;MAKgD,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAN,K  
AAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAN,KAAM,0C;K;mDAGzD,iB;MAK2C,OAAM,WAAN,KAAM,q  
C;K;mDAEjD,iB;MAK4C,OAAM,aAN,KAAM,qC;K;mDAEID,iB;MAS8C,OAAM,aAN,KAAM,qC;K;mDAG  
pD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAN,KAAM,qC;K;mDAEID,iB;MA  
S8C,OAAM,aAN,KAAM,qC;K;iDAGpD,iB;MAKyC,OAAM,WAAN,KAAM,mC;K;iDAE/C,iB;MAK0C,OAA  
M,aAN,KAAM,mC;K;iDAEhD,iB;MAS4C,OAAM,aAN,KAAM,mC;K;gDAGID,iB;MAKwC,OAAM,WAAN,  
KAAM,kC;K;gDAE9C,iB;MAKyC,OAAM,aAN,KAAM,kC;K;gDAE/C,iB;MAS2C,OAAM,aAN,KAAM,kC;  
K;iDAEjD,iB;;QAY4C,OACxC,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,sCAAmC,K  
AAAnC,OAAzB,EAA sE,CAAtE,C;;UAHkC,O;;K;0DAM5C,iB;;QAeqD,OACjD,cAAc,KAAAd,EAAiC,IAAjC,C;;Q

ACF,+C;UACE,MAAM,6BAAyB,0CAAuC,KAAvC,OAAzB,EAA0E,CAA1E,C;;UAH2C,O;;K;uDAMrD,iB;;QA  
WmD,OAC/C,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UAFiD,OAG/C,I;;UAH+C,O;;K;gEAMnD,iB;;QAO4D,  
OACxD,cAAc,KAAAd,EAAiC,IAAjC,C;;QACF,+C;UAF0D,OAGxD,I;;UAHwD,O;;K;;IA1YhE,0C;MAAA,yC;Q  
AAA,wB;;MAAA,kC;K;oCAmZA,Y;MAC6C,kBAAY,YAAD,aAAX,EAPaK,eAAS,QAAT,GAAqB,CAoA1B,C;K  
;qCAE7C,iB;MAiBW,Q;MATH,IAAA,IAAK,aAAL,C;QACI,IAAI,KAAM,WAAN,IAAqB,IAAK,WAAL,KAAkB  
,KAAM,WAAXB,gBAAoC,CAA7D,C;UACI,OAAO,I;;UAEP,MAAM,gCAAyB,2EAAzB,C;WAEd,IAAA,KAAM  
,aAAN,C;QAAsB,OAAO,K;MAI7B,KAXb0C,eAAS,QAAT,GAAqB,CAwb/D,OAA0B,KAXbGB,WAAS,QAAT,G  
AAqB,CAwb/D,E;QACI,aAAa,IAAK,QAAL,KAAa,KAAM,QAANB,C;QAET,uB;UACI,ICAA0B,MAA1B,C;;UA  
EA,kCAA2B,MAA3B,C;aAGZ,IAAA,IAAK,eAAL,C;QACI,mCAAqB,IAAK,QAA1B,EAAiC,KAAM,QAAvC,C;  
;QAEA,mCAAqB,KAAM,QAA3B,EAAkC,IAAK,QAAvC,C;MABR,W;K;gDAiBJ,kC;MAGW,Q;MAFP,kBAaKB  
,cAAc,UAAAd,C;MACIB,mBAAmB,eAAa,WAAb,C;MACZ,IAAI,8EAAc,mBAAtC,CAAJ,C;QACH,yBAAYB,o  
BAAa,cAAc,WAAd,CAAb,C;QACzB,uBAAGB,cAAc,YAAAd,MAA8B,kBAA9B,CAAhB,C;;QAEA,wBAA8B,W  
AAb,YAAa,yBAAsB,UAAtB,CAA9B,C;;MAJJ,W;K;sCAQJ,iB;MAMuD,wBAAS,KAAD,aAAR,C;K;uCAEvD,iB  
;MAQe,UAUJ,M;MAXP,IAAI,iBAAJ,C;QAEQ,cAAS,CAAT,C;UAAc,MAAM,gCAAyB,mEAAzB,C;aACpB,YA  
AQ,CAAR,C;UAAa,W;;UACL,OAAC,IAAD,a;QAHZ,W;;MAMJ,IAAI,UAAS,CAAb,C;QAAgB,OAAO,qC;MAE  
vB,YAAAY,Y;MACZ,aAAa,mCAAQ,KAAR,E;MACN,IAAI,kBAAJ,C;QACH,IAAI,yEAAJ,C;UAEI,yBAAGB,MA  
AhB,C;;UAEA,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;YACI,mCAA0B,MAA1B,C;;YAEA,aAAa,cAAc,K  
AAAd,C;YACb,eAAe,eAAQ,cAAc,MAAd,CAAR,C;YACf,mBAAmB,oCAAS,KAAT,E;YACnB,kBAaKB,iBAaE,c  
AAc,sCAAW,KAAx,EAAd,CAAf,C;YACIB,IAAI,4CAAe,KAAf,IAAwB,MAAxB,KAAkC,gBAAGB,YAAhB,gB  
AAgC,CAAtE,C;cACI,0BAA6B,WAAZ,WAAAY,EAAS,8BAAa,UAAb,CAAT,CAA7B,C;;cAEA,SAAI,YAAM,W  
AAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAANB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;;QAK3D,  
IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;UACI,0BAAwB,WAAP,MAAO,EAAS,8BAAa,UAAb,CAAT,CAA  
xB,C;;UAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAANB,IAA0B,CAA9B,GAAiC,yC  
AAjC,GAA+C,qD;;;MAvBvD,a;K;uCA4BJ,iB;MASI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,K  
AA3B,C;QACI,OAAO,mBAAM,QAAN,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,O  
AAc,aAAP,MAAO,EAAW,IAAX,C;K;qCAGIB,iB;MAQe,Q;MADX,IAAI,UAAS,CAAb,C;QAEQ,sB;UAAgB,gD  
;aChB,sB;UAAgB,4D;;UACR,MAAM,gCAAyB,4DAAzB,C;QAHIB,W;;MAMJ,IAAI,kBAAJ,C;QACI,OAAO,g  
BAAGB,qCAAQ,KAAR,EAhB,C;;QAEP,IAAI,iBAAJ,C;UACI,OAAO,mBAAa,WAAN,KAAM,CAAb,C;QAEX  
,aAAa,qCAAQ,KAAR,E;QAEb,IAAI,kEAAgC,mBAAhC,CAAJ,C;UACI,UAAU,cAAc,sBAAS,oCAAS,KAAT,E  
AAT,CAAd,0BAA0C,KAA1C,E;UACV,OAAO,gBAAGB,cAAc,MAAd,MAAwB,GAAxB,CAAhB,C;;QAEX,OA  
AO,iBAAiB,MAAjB,C;;K;qCAIf,iB;MAOI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAAvB,IA  
AgC,aAAY,CAAhD,C;QACI,OAAO,iBAAI,QA AJ,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;  
MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;oCAGIB,iB;MAEI,kBAaKB,SAAM,IAAK,cAAX,EAawB,KA  
AM,cAA9B,C;MACIB,OAAO,IAAK,kBAAS,WAAT,CAAL,GAA6B,KAAM,kBAAS,WAAT,C;K;oCAG9C,Y;M  
ACmC,oCAAW,C;K;oCAE9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,+BAAY,yCAAS,WAArB,KAAiC,  
wBAAY,qDAAa,WAAzB,C;K;kCAEpE,Y;MACiC,QAAC,iB;K;yFAGC,Y;MAAQ,OAAI,iBAAJ,GAAMB,IAAD,  
aAAIB,GAA6B,I;K;yCAExE,iB;MACI,kBAaKB,IAAK,WAAL,KAAkB,KAAM,WAAXB,C;MACIB,IAAI,yBAAc  
,CAAd,IAAMB,CAAA,WAAAY,QAAZ,GAawB,CAAXB,MAA6B,CAApD,C;QACI,OAAO,IAAK,WAAS,iBAAU  
,KAAM,WAAhB,C;MAEZB,QAAQ,CArmBsC,eAAS,QAAT,GAAqB,CAqmB3D,KAAyB,KArmBa,WAAS,QAA  
T,GAAqB,CAqmB3D,K;MACR,OAAW,iBAAJ,GAakB,CAAC,CAAD,IAAIB,GAA0B,C;K;uHAMrC,kB;MAeI,  
OAAO,OAAO,gBAAP,EAAoB,mBAAPB,EAAoC,qBAAPC,EAAcD,qBAAtD,EAAwE,yBAAXE,C;K;uHAGX,kB  
;MAcI,OAAO,OAAO,iBAAP,EAAqB,qBAArB,EAAuC,qBAAvC,EAAyD,yBAAZD,C;K;uHAGX,kB;MAAI,OAA  
O,OAAO,mBAAP,EAAuB,qBAAvB,EAAyC,yBAAZC,C;K;uHAGX,kB;MAYI,OAAO,OAAO,mBAAP,EAAuB,y  
BAAvB,C;K;0FAKP,Y;MAAQ,OAAI,iBAAJ,GAakB,CAAI B,GAA0B,6CAAe,EAAf,EAAMB,Q;K;4FAIvD,Y;M  
AAQ,OAAI,iBAAJ,GAakB,CAAI B,GAA0B,+CAAiB,EAAjB,EAAqB,Q;K;4FAIvD,Y;MAAQ,OAAI,iBAAJ,GA  
AkB,CAAI B,GAA0B,+CAAiB,EAAjB,EAAqB,Q;K;gGAIvD,Y;MACI,sB;QADI,OACY,C;WACHB,wB;QAFI,OA  
EY,cAAc,wCAAQ,IAAR,EAAd,CAA6B,Q;;QAFzC,OAGK,wCAAQ,UAAR,EAAuB,Q;K;0CAMxC,gB;MAQiB,  
UAAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;QAA4B,SAAP,wCAAO,kB;WAC5B,iBAAA,qDAAa,W

AAb,E;QAAGC,SAAP,wCAAO,kB;;QAG5B,6BAAoB,YAAM,WAA1B,EAAcC,kBAAtC,EAAmD,IAAnD,C;;MALR,a;K;wCAUJ,gB;MAUiB,UAAAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;;WACA,iBAAA,qDAAa,WAAAb,E;;;QACQ,+BAAoB,YAApB,EAA2B,kBAA3B,EAAwC,IAAxC,C;MAHZ,a;K;uCAOJ,gB;MAUI,OAAa,WAAb,oBAAO,IAAP,CAAA,4BAAyD,Q;K;kFAMhD,Y;MAAQ,6D;K;mfAMP,Y;MAAQ,8D;K;qFAMN,Y;MAAQ,gE;K;qFAMR,Y;MAAQ,gE;K;0FAMH,Y;MAAQ,qE;K;0FAMR,Y;MAAQ,qE;K;yFAMT,Y;MAAQ,oE;K;uFASrC,Y;MAAQ,2D;K;wFAQR,Y;MAAQ,4D;K;0FAQR,Y;MAAQ,8D;K;0FAQR,Y;MAAQ,8D;K;+FAQR,Y;MACI,OAAW,uBAAgB,eAApB,GAAgC,YAAhC,GAA2C,4D;K;+FAAtD,Y;MAAQ,mE;K;8FAYR,Y;MAEW,Q;MADP,YAAY,Y;MAER,uB;QA Ae,Y;WACf,8C;;WACA,+C;;QACQ,qBAAc,KAAAd,C;MAJZ,W;K;2CAUR,Y;MAUuC,8B;K;4CAEvC,Y;MAUwC,+B;K;kCAExC,Y;MAuBwC,Q;MAAA,sB;MACpC,qB;QAD8B,OACxB,I;WACN,iBAAAY,yCAAS,WAAT,E;QAF8B,OAET,U;WACrB,iBAAA,qDAAa,WAAAb,E;QAH8B,OAGL,W;;QAErB,iBAAiB,iB;Q4H7iBF,gBAAhB,sB;Q5H+iBK,e;UAAgB,yBAAO,EAAP,C;QACF,YAAAd,kB;QAvSD,WAAO,iB;QAAP,YAAoB,oB;QAAPB,cAAoC,sB;QAAPC,cAAsD,sB;QAAtD,kBAAwE,0B;QA+S/D,0B;QAPJ,cAAc,iB;QACd,eAAe,UAA S,C;QACxB,iBAAiB,YAAW,C;QAC5B,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C;QAChD,iBAAiB,C;QACjB,IAAI,OAAJ,C;UACI,yBAAO,IAAP,CAAA,gBAAO,GAAP,C;UACb,+B;;QAEJ,IAAI,aAAa,YAAY,cAAc,UAA1B,C AAb,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,KAAP,CAAc,gBAAO,G AAP,C;;QAEIB,IAAI,eAAe,eAAe,YAAY,OAA3B,CAAf,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAA O,EAAP,C;UACtB,yBAAO,OAAP,CAAgB,gBAAO,GAAP,C;;QAEpB,IAAI,UAAJ,C;UACI,IAAI,6DAAe,CAAn B,C;YAAsB,yBAAO,EAAP,C;UAEIB,gBAAW,CAAX,IAAgB,OAAhB,IAA2B,QAA3B,IAAuC,UAAvC,C;YACI ,mCAAiB,OAAjB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C,GAA1C,EAA2D,KAA3D,C;eACJ,mBAAe,OAAf,C ;YACI,mCAAiB,cAAc,OAAd,IAAjB,EAA0C,cAAc,OAAd,IAA1C,EAAmE,CAAnE,EAAeE,IAAtE,EAaWf,KA AxF,C;eACJ,mBAAe,IAAf,C;YACI,mCAAiB,cAAc,IAAd,IAAjB,EAAcC,cAAc,IAAd,IAAtC,EAA2D,CAA3D,E AA8D,IAA9D,EAAGf,KAAhF,C;;YAEA,yBAAO,WAAP,CAAoB,gBAAO,IAAP,C;;QAGhC,IAAI,cAAc,aAAa,C AA/B,C;UAAkC,yBAAO,CAAP,EAAU,EA AV,CA Ae,gBAAO,EAAP,C;QAvC/B,OQ52B3B,SoHoUqC,W;;K;4C 5HqlB5C,yE;MACI,yBAAO,KAAP,C;MACA,IAAI,eAAc,CAAIB,C;QACI,yBAAO,EAAP,C;QACA,iBA AuC,W AAAtB,UAAW,WAAW,EAAS,cAAT,EAAYB,EAazB,C;QACR,sB;;UuB/0BzB,Q;UAAA,OAAQ,WAAR,evB+0Bc ,UuB/0Bd,CAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IvB80BiD,UuB90BnC,YvB80BU,UuB90BV ,YAAK,KAAL,EvB80BmC,MAAM,EuB90BvD,C;cACI,qBAAO,K;cAAP,uB;;UAGR,qBAAO,E;;;QvB00BC,oB AAoB,qBAAuC,CAAvC,I;QAEhB,KAAC,SAAD,IAAc,gBAAgB,CAA9B,C;UAAmC,8BAAAY,UAAZ,EAaWb,C AAxB,EAA2B,aAA3B,C;;UAC3B,8BAAAY,UAAZ,EAaWb,CAAxB,EAA2B,CAAC,CAAC,gBAAgB,CAAhB,IA AD,IAAsB,CAAtB,IAAD,IAA4B,CAA5B,IAA3B,C;;MAGhB,yBAAO,IAAP,C;K;0CAGJ,0B;MAGbWc,wB;QAA A,WAAgB,C;MKv+BxD,IAAI,ELw+BQ,YAAY,CKx+BpB,CAAJ,C;QACI,cLu+ByB,oD;QKt+BzB,MAAM,gCA AyB,OAAQ,WAAjC,C;;MLu+BN,aAAa,sBAAS,IAAT,C;MACb,IAAW,WAAP,MAAO,CAAX,C;QAAyB,OAAO ,MAAO,W;MACvC,OAAO,sBAAsB,MAAtB,EAAuC,eAAT,QAAS,EAAa,EAAb,CAAvC,IAAgE,UAAAL,IAAK,C ;K;qCAI3E,Y;M4H3nBuB,gBAAhB,sB;M5HyOBH,IAAI,iBAAJ,C;QAakB,yBAAO,EAAP,C;MACIB,yBAAO,IA AP,C;MAC4B,YAAAd,kB;MAjXP,YAAO,kB;MAAP,cAAqB,sB;MAArB,cAAuC,sB;MAAvC,kBAAyD,0B;MAkX 5D,cACY,K;MACZ,IAAI,iBAAJ,C;QAEI,wB;;MAEJ,eAAe,oB;MACf,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C; MACHD,iBAAiB,YAAW,CAAX,KAAiB,cAAc,QAA/B,C;MACjB,IAAI,QAAJ,C;QACI,yBAAO,OAAP,CAAc,g BAAO,EAAP,C;;MAEIB,IAAI,UAAJ,C;QACI,yBAAO,OAAP,CAAgB,gBAAO,EAAP,C;;MAEpB,IAAI,eAAe,C AAC,QAAD,IAAa,CAAC,UAA7B,CAAJ,C;QACI,mCAAiB,OAAjB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C, GAA1C,EAA2D,IAA3D,C;;MApBuB,OQ58B5B,SoHoUqC,W;K;;kC5H5YhD,Y;MAAA,c;MAuBiD,2D;MAvBj D,a;K;gCAAA,iB;MAAA,2IAuBiD,gDAvBjD,G;K;IAkjCA,qC;MAIW,Q;MAAA,IAAI,6DAAJ,C;QACH,uBAAg B,4BAAiC,oBAAL,SAAK,CAAjC,EAA2C,IAA3C,yCAAhB,C;;QAES,oBAAT,8BAAS,EAaW,IAAX,C;MAHb, W;K;IAMJ,uC;MAII,kBAAkB,4BAA4B,SAA5B,0CAAiE,IAAJE,C;MACIB,IAAa,WAAD,aAAR,yDAAsB,WAAt B,CAAJ,C;QACI,OAAO,gBAAgB,4BAA4B,SAA5B,EAakC,IAAI,C,yCAAhB,C;;QAEp,aAAa,sBAAoB,SAAPB, EAA0B,IAA1B,0C;QACb,OAAO,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;IAIf,uC;MAaW,Q;M AHP,gBAAgB,oBAAoB,SAAPB,EAA0B,IAA1B,yC;MK3jChB,IAAI,CL4jCI,CAAW,QAAV,SAAU,CK5jCnB,C; QACI,cL2jC0B,+B;QK1jC1B,MAAM,gCAAYB,OAAQ,WAAjC,C;;ML2jCV,YAAsB,YAAV,SAAU,C;MACf,IA AI,sEAAqB,SAArB,CAAJ,C;QACH,uBAAgB,KAAhB,C;;QAEA,aAAwE,YAA3D,oBAAoB,SAAPB,EAA0B,IA

A1B,0CAA2D,C;QACxE,kCAA2B,MAA3B,C;;MAJJ,W;K;IAgBuB,oC;MAAQ,oE;K;IAOP,sC;MAAQ,sE;K;IAW  
N,sC;MAAQ,sE;K;IAQV,qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQX,qC;MAAQ,qE;  
K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQhB,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;M  
AAQ,kE;K;IAQX,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQb,8B;MAAQ,8D;K;IAOP  
,gC;MAAQ,gE;K;IAWN,gC;MAAQ,gE;K;IAQZ,6B;MAAQ,6D;K;IAOP,+B;MAAQ,+D;K;IAWN,+B;MAAQ,+D;  
K;yEAG/B,+B;MAIqE,8BAAW,SAAX,C;K;2EAERe,+B;MAUwE,8BAAW,SAAX,C;K;IAIxE,yC;MACI,aAAa,K  
AAM,O;MACnB,IAAI,WAAU,CAAd,C;QAAiB,MAAM,gCAAYB,qBAAZB,C;MACvB,YAAY,C;MACZ,aAAa,g  
CAAS,K;MACtB,qBAAqB,U;MACrB,QAAM,iBAAM,KAAN,CAAN,C;aACI,E;aAAA,E;UAAy,qB;UAAZ,K;;M  
AEJ,cAAc,QAAQ,C;MACtB,iBAAiB,WAAiB,aAN,KAAM,EAAW,EAAX,C;MAE9B,cAAU,KAAY,C;QACI,  
MAAM,gCAAYB,eAAzB,C;WACV,qBAAM,KAAN,MAAGB,EAHbB,C;QACI,IAAI,mCAAW,MAAf,C;UAAuB,  
MAAM,+B;QAC7B,sBAAsB,K;QACtB,sBAAsB,K;QACtB,eAA8B,I;QAC9B,OAAO,QAAQ,MAAf,C;UACI,IAA  
I,iBAAM,KAAN,MAAGB,EAAPB,C;YACI,IAAI,mBAAM,mCAAW,MAAIC,C;cAA0C,MAAM,+B;YACHD,kB  
AAkB,I;YACIB,Q;;UAekB,iBAAE,K;UA+EjD,QAHgC,U;UAIhC,Y;YAAO,eAhFqB,KAgFjB,O;YAAJ,S;cAAc,S  
AAU,YAhFH,KAgFG,YAAK,CAAL,E;cAAV,OAhFqC,CAAM,kBAAK,EAAL,CAAN,qCAAKB,2C;;;YAgFnC,a  
;;UAhF7B,gBAAGB,KkBlICgE,WIB8pClF,UkB9pCkF,ElBmqCrF,CkbnqCqF,C;UIBmlChF,IAAI,SwBziCgC,YA  
AU,CxByiC9C,C;YAAyB,MAAM,+B;UAC/B,gBAAS,SAAU,OAAAnB,I;UACqB,cAAU,K;UuB5sCpC,U;UAAA,I  
AAI,WAAS,CAAT,IAAc,WAAS,iBvB4sCP,KuB5sCO,CAA3B,C;YAAA,SvB4sCoB,KuB5sCkB,YAAI,OAAJ,C;;  
YvB4sCO,MAAM,gCAAYB,qCAAZB,C;;UAA9C,qB;UACA,qB;UACA,WAAW,sBAAsB,QAAtB,EAAGC,eAAh  
C,C;UACX,IAAI,YAAY,IAAZ,IAAoB,yBAAY,IAAZ,MAAxB,C;YAA0C,MAAM,gCAAYB,yCAAZB,C;UACHD,  
WAAW,I;UACX,eAAyB,WAAV,SAAU,EAAQ,EAAR,C;UACzB,IAAI,+CAAGC,WAAW,CAA/C,C;YACI,YAA  
Y,SkB5ICgE,WIB4IC5C,CkB5IC4C,EIB4ICzC,QkB5ICyC,C;YIB6IC5E,4BAA2C,aAAjC,0BAA0B,KAA1B,CAAi  
C,EAAW,IAAX,CAA3C,C;YACA,4BAAMd,aAAX,SAA9B,SkBjmCmD,WIBimC/B,QkBjmC+B,CIBimCrB,CA  
AW,EAAW,IAAX,CAAnD,C;;YAEA,4BAA+C,aAARc,0BAA0B,SAA1B,CAAqC,EAAW,IAAX,CAA/C,C;;aAI  
Z,c;QACI,MAAM,+B;;QACV,IAAM,cAAN,KAAM,EAAC,KAAd,EAAqB,cAARb,EAAqC,CAARc,ESnzCH,MA  
AO,KTmzCmD,SAAS,KAAT,ISnzCnD,ETmzCmE,cAAe,OSnzClF,CTmzCJ,EAA4G,IAA5G,CAAN,C;UACI,SA  
AS,gCAAS,S;;UAiIB,iBAA8B,I;UAC9B,iBAAiB,K;UACjB,kBAAkB,CAAC,O;UACnB,IAAI,WAAW,iBAAM,K  
AAN,MAAGB,EAA3B,IAAwC,QAAN,KAAM,CAAN,KAAGB,EAAtD,C;YACI,cAAc,I;YACd,IAAI,oCAAW,uB  
AAX,EAAW,MAAX,CAAJ,C;cAAyB,MAAM,gCAAYB,eAAzB,C;;UAEnc,OAAO,QAAQ,MAAf,C;YACI,IAAI,  
cAAc,WAAIB,C;cA8CZ,UA7CwC,K;cA8Cx,C;Y;gBAAO,mBA9CiB,KA8Cb,O;gBAAJ,W;kBAAC,SA9C4B,UA8  
CIB,YA9CP,KA8CO,YAAK,GAAL,EA9CkB,MAAM,E;;;gB8Ccd,iB;;cA9CzB,QA+CT,G;;YA7CK,aAAa,I;YA  
CS,mBAAE,K;YA0CjD,UAHgC,Y;YAIhC,Y;cAAO,mBA3CqB,KA2CjB,O;cAAJ,W;gBAAC,WAAU,YA3CH,KA  
2CG,YAAK,GAAL,E;gBAAV,SA3CqC,CAAM,kBAAK,EAAL,CAAN,uCAAKB,oBAAM,E;;;cA2CzC,iB;;YA3C  
7B,kBAAGB,KkBvnCgE,WIB8pClF,YkB9pCkF,ElBmqCrF,GkbnqCqF,C;YIBwnChF,IAAI,WwB9kCgC,YAAU,C  
xB8kC9C,C;cAAyB,MAAM,+B;YAC/B,gBAAS,WAAU,OAAAnB,I;YACqB,mBAAE,K;YAUChD,UAHgC,Y;YAI  
hC,Y;cAAO,mBAxCoB,KAwChB,O;cAAJ,W;gBAAC,WAAU,YAxCJ,KAwCI,YAAK,GAAL,E;gBAAV,SAxCoC  
,CAAM,kBAAK,GAAL,CAAN,mC;;;cAwChB,iB;;YAx7B,eAAe,KkB1nCiE,WIB8pClF,YkB9pCkF,ElBmqCrF,  
GkbnqCqF,C;YIB2nChF,gBAAS,QAAS,OAAIB,I;YACA,aAAW,wBAAwB,QAAXB,C;YACX,IAAI,cAAY,IAAZ  
,IAAoB,2BAAY,MAAZ,MAAxB,C;cAA0C,MAAM,gCAAYB,yCAAZB,C;YACHD,aAAW,M;YACX,iBAAYB,W  
AAV,WAAU,EAAQ,EAAR,C;YACzB,IAAI,aAAW,CAAf,C;cACI,cAAY,WkBjOCgE,WIBioC5C,CkBjOC4C,ElBi  
oCzC,UkBjOCyC,C;cIBkoC5E,4BAAYB,aAAT,OAAO,OAAM,CAAS,EAAW,MAAX,CAAZB,C;cACA,4BAAMd  
,aAAX,SAA9B,WkBtoCmD,WIBsoC/B,UkBtoC+B,CIBsoCrB,CAAW,EAAW,MAAX,CAAnD,C;cACA,IAAI,QA  
AQ,MAAZ,C;gBAAoB,MAAM,gCAAYB,mCAAZB,C;;cAE1B,4BAA6B,aAAT,OAAV,WAAU,CAAS,EAAW,M  
AAX,CAA7B,C;;;MAKhB,OAAW,UAAJ,GAAiB,MAAD,aAAhB,GAA6B,M;K;IAIxC,0C;MACI,aAAa,KAAM,  
O;MACnB,iBAAiB,C;MACjB,IAAI,SAAS,CAAT,IAAc,YAAY,IAAZ,mBAAM,CAAN,EAAlB,C;QAAoC,+B;;M  
ACHC,YAAC,SAAS,UAAT,IAAD,IAAwB,E;MAAxB,S;QAA4D,gBAA7B,yBAAkB,iBAAN,KAAM,CAAIB,C;Q  
AA6B,c;;UW8ShD,U;UADhB,IAAI,wCAAsB,mBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,6B;UAAhB,OAAg  
B,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CX9S4C,CAAa,kBAAK,EAAL,CAAb,oCW8SjC,OX9SiC,EW8ShD,C;c  
AAyB,aAAO,K;cAAP,e;;UAC/C,aAAO,I;;QX/SyD,iB;;MAAhE,S;QAEI,OAAW,iBAAM,CAAN,MAAY,EAHh

B,sD;;MAGX,OAAiB,WAAN,KAAM,EAAW,GAAX,CAAV,GAAyC,OAAR,QAAN,KAAM,EAAK,CAAL,CAAQ,CAAzC,GAA6D,OAAN,KAAM,C;K;IAKxE,0D;MAII,QAHgC,U;MAIhC,OAAO,IAAI,gBAAJ,IAJqC,SAIvB,CAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAyC,a;;MAJzC,OkB9pC4F,oBIB8pC1F,UkB9pCkF,ElBmqCrF,CkbnqCqF,C;K;IIBgqChG,qD;MACI,QAAQ,U;MACR,OAAO,IAAI,gBAAJ,IAAc,UAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAyC,a;;MACzC,OAAO,C;K;I;AmBX,8B;MAA+C,qCAAQ,OAAR,E;K;IAC/C,+B;MAAgD,2CAAS,OAAT,E;K;IAEHd,sC;MAAiD,oBAAS,sBAAGb,CAAhB,CAAT,C;K;IACjD,wC;MAAmD,oBAAU,uBAAiB,CAAjB,CAAD,yBAAuB,CAAvB,EAAT,C;K;IACnD,oD;MAAoE,oBAAU,sBAAGb,CAAhB,CAAD,yBAAsB,iBAAtB,EAAT,C;K;IACpE,0C;MACI,IAAI,sEAAqB,SAArB,CAAJ,C;QAAA,OACI,gBAAgB,KAAhB,C;;QADJ,OAGI,iBAAiB,cAAc,KAAAd,CAAjB,C;;K;IAGR,4C;MACI,IAAI,kEAAgC,mBAAhC,CAAJ,C;QAAA,OACI,gBAAgB,cAAc,MAAd,CAAhB,C;;QADJ,OAGI,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;I6Mt4CR,8B;MAEgD,QAAM,SAAN,M;aAC5C,a;UAD4C,OACHb,I;aAC5B,c;UAF4C,OAEf,I;aAC7B,c;UAH4C,OAGf,I;aAC7B,S;UAJ4C,OAIpB,G;aACxB,S;UAL4C,OAKpB,G;aACxB,O;UAN4C,OAMtB,G;aACtB,M;UAP4C,OAovB,G;;UxMuEwB,MAAM,6BAA8B,CwMtEnE,mBAAGb,SxMsEmD,YAA9B,C;;K;IwMnEvD,4C;MACwE,QAAM,SAAN,C;aACpE,I;UADoE,6C;aAEpE,I;UAFoE,8C;aAGpE,I;UAHoE,8C;aAIpE,G;UAJoE,yC;aAKpE,G;UALoE,yC;aAMpE,G;UANoE,uC;aAOpE,G;UAPoE,sC;;UAQ5D,MAAM,gCAAYb,uCAAoC,SAAT7D,C;;K;IAGIB,yD;MAGQ,KAAC,eAAD,C;QAEQ,IADE,OACF,Q;UAHZ,sC;;UAIoB,MAAM,gCAAYb,4EAAqD,OAARd,CAAzB,C;;QAIIB,QAAM,OAAN,C;eACI,E;YATZ,uC;eAUyE,YAVZ,yC;eAWY,E;YAXZ,yC;;YAYoB,MAAM,gCAAYb,yDAaKc,OAAlC,CAAzB,C;;K;IC5F9B,4B;K;;MC8FI,kC;;IAlEA,gC;MAAA,oC;K;6CAUI,Y;MAAwC,OAAA,iCAAoB,U;K;8CAC5D,Y;MAAkC,OAAA,iCAAoB,W;K;IAiBrB,qD;MAAqB,8B;K;8DACID,Y;MAAsC,OAAA,iCAAoB,qBAAy,IAAZ,C;K;+DAC1D,oB;MAAuD,OAAA,iCAAoB,uBAAc,IAAd,EAAoB,QAAPb,C;K;gEAC3E,oB;MAAwD,OAAA,iCAAoB,uBAAc,IAAd,EAAqB,QAAD,aAApB,C;K;gEAC5E,Y;MAAuC,QAAC,iBAAa,a;K;mEACrD,Y;MAA0C,OAAA,iBAAa,a;K;iEAEvD,iB;MACI,IAAI,yDAAJ,C;QACI,MAAM,gCAAYb,sFAAmF,IAAnF,aAA6F,KAAtH,C;MACV,OAAO,IAAK,eAAM,KAAN,C;K;iEAGhB,iB;MAW4D,OAAA,iCAAoB,2BAAkB,IAAIB,EAAwB,KAAxB,C;K;qEAEhF,iB;MAQI,OAAC,mBAAO,KAAP,CAAc,iBAAU,gCAAS,KAAAnB,C;K;;4DAjDvB,Y;MAAA,OAesD,gEafD,M;K;4DAAA,Y;MAAA,c;MAesD,gE;MAftD,a;K;0DAAA,iB;MAAA,2IAesD,0DAftD,G;K;;IABJ,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAKEA,gC;MAAA,oC;K;;IAAA,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;;qCA2BA,oB;MAWyD,4BAAiB,IAAjB,EAuB,QAavB,C;K;sCAEzD,oB;MAW+D,wBAAM,QAAD,aAAL,C;K;sCAG/D,Y;MAMqC,QAAC,iBAAa,a;K;yCAEnD,Y;MAMwC,OAAA,iBAAa,a;K;;gDAUrD,oB;MAAkF,wBAAM,QAAD,aAAL,C;K;mDAmBIF,iB;MAyI,OAAC,mBAAO,KAAP,CAAc,iBAAU,gCAAS,KAAAnB,C;K;;IAYO,4C;MAAC,gB;MAAoB,4B;K;4CAC/C,Y;MAAsC,OAAA,SAAK,aAAL,cAAoB,eAAPb,C;K;6CAEtC,oB;MAAkD,4BAAiB,SAAjB,EAuB,4BAAa,QAAb,CAAvB,C;K;;ICxMV,sC;MAAC,gB;K;IAOf,4E;MAAC,4B;MAA6B,8B;MAAgD,sB;K;+DACpG,Y;MAAsC,OAAI,aAAO,aAAX,GAA0B,aAAD,aAAzB,GAAsE,aAA/B,iBAAW,OAAx,UAAoB,gBAApB,CAA+B,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;gEAC5E,oB;MAA4D,+CAAa,gBAAb,EAAwB,iBAAxB,EAAoC,0BAAS,QAAT,CAApC,C;K;kEAC5D,iB;MAQa,Q;MAPT,IAAI,8DAA0B,QAAA,IAAK,aAAL,EAAM,KAAM,aAAzB,CAA9B,C;QACI,MAAM,gCAAYb,sFAAmF,IAAnF,aAA6F,KAAtH,C;MAMV,IAAI,EAAK,OAAL,IAAK,SAAL,wBAAE,KAAM,SAArB,aAA+B,IAAK,SAAO,aAA/C,C;QA A6D,OAAO,gCAAS,K;MAC7E,iBAAiB,IAAK,SAAL,cAAc,KAAM,SAAPb,C;MACjB,oBAAuD,aAAlC,IAAK,YAAL,UAAiB,KAAM,YAAvB,CAAKC,EAAW,iBAAW,KAAtB,C;MAEvD,OAAO,CAAI,6CAAKB,UAAD,aAAjB,QA AJ,IAAkC,gCAAS,KAA3C,GAAqD,0BAAGb,UAAhB,C;K;2DAGhE,iB;MACqE,Q;MAAJE,oEAAyB,OA AA,IAAK,aAAL,EAAM,KAAM,aAAzB,CAAzB,KAAGe,CAAC,0BAAO,KAAP,CAAD,wBAAkB,gCAAS,KA A3B,QA AhE,C;K;8EAEJ,Y;MACI,IAAI,aAAO,aAAX,C;QAAyB,OAAO,a;MACHC,WA AW,iBAAW,K;MACTb,IAAI,kEAAJ,C;QACI,OAAiB,aAAV,gBAAU,EAAW,IAAX,CAAV,aAA6B,aAA7B,C;;MAEX,YAAY,mEAAmD,IAAnD,C;MACZ,sBAAsB,qBAAY,KAAZ,C;MACTb,mBAAMb,wBAAY,KAAZ,C;MAEL,YAAP,a;MhN+oBJ,o BAAO,oB;MAAP,wBAAuB,0B;MgN9oBtB,mBAAMb,oBAAoB,OAAPb,I;MACnB,qBAAqB,oBAAoB,OAAPb, I;MAFzB,OAKkB,gCAAb,ahNAiC,WgNAD,chNAC,2BgNAjC,CAAD,ahN0CoC,agNzC3B,wCAAKB,YAAIB,Eh NyC2B,4BgN1CpC,chN6D+B,agN3DvB,ahN2DuB,uBgN7D/B,C;K;6DAOR,Y;MAA+B,OAAA,gCAAoB,W;K;6 DAEnD,Y;MAAkC,yBAAE,gBAAf,cAA0C,UAAhB,iBAAW,KA AK,CAA1C,WAA0D,aAA1D,WAAqE,gCAArE,WAA6F,iBAA7F,M;K;;+CAGtC,Y;MAA6C,+CAAa,WAAb,EAAqB,IAArB,EAA2B,gCAAS,KAApC,C;K;;IAW

H,wC;MAAC,gB;K;IAQf,gF;MAAC,4B;MAA+B,8B;MAAkD,sB;K;mEAC1G,Y;MAAsC,OAAGC,aAA/B,iBAAW,OAAX,GAAoB,gBAAW,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;oEACtC,oB;MAA4D,mDAAe,gBAAf,EAA0B,iBAA1B,EAA5C,0BAAS,QAAT,CAAtC,C;K;sEAE5D,iB;MAIa,Q;MAHT,IAAI,kEAA4B,QAAA,IAAK,aAAL,EAAmB,KAAM,aAAzB,CAAhC,C;QACI,MAAM,gCAAYB,sFAAmF,IAAnF,aAA6F,KAAtH,C;MAEV,IAAI,EAAK,OAAL,IAAK,SAAL,wBAAe,KAAM,SAArB,aAA+B,IAAK,SAAO,aAA/C,C;QAA6D,OAAO,gCAAS,K;MAC7E,iBAAiB,IAAK,SAAL,cAAc,KAAM,SAAPB,C;MACjB,oBAAuD,aAAIC,IAAK,YAAL,GAAiB,KAAM,YAAW,EAAW,iBAAW,KAAtB,C;MACvD,OAAO,CAAI,6CAAKB,UAAD,aAAjB,QAAJ,IAAKC,gCAAS,KAA3C,GAAqD,0BAAGB,UAAhB,C;K;+DAGhE,iB;MAC8E,Q;MAA1E,OAAO,iEAA2B,OAAA,IAAK,aAALL,EAAmB,KAAM,aAAzB,CAA3B,KAAkE,CAAC,0BAAO,KAAP,CAAD,wBAAkB,gCAAS,KAA3B,QAAIE,C;K;iEAGX,Y;MACI,OAAkB,aAAV,gBAAU,EAAW,iBAAW,KAAtB,CAAV,aAAwC,aAAxC,CAAGD,W;K;iEAG5D,Y;MAAkC,2BAAiB,gBAAjB,GAA4C,UAAhB,iBAAW,KAAK,CAA5C,WAA4D,aAA5D,UAAqE,iBAARe,M;K;;iDAGtC,Y;MAA6C,mDAAe,WAAf,EAAuB,IAAvB,EAA6B,gCAAS,KAAtC,C;K;;IAGjD,0B;MAG8B,yE;MAC1B,mB;K;oCAEA,Y;MAA4B,qB;K;iDAE5B,oB;MAWc,Q;MADV,gBAAgB,QAAS,gBAAO,SAAP,C;MACf,IAAI,gDAA+B,4CAAnC,C;QAEN,iBAAiB,mBAAU,SAAV,C;QACjB,IAAI,mBAAy,SAAZ,gBAAyB,CAAzB,IAA8B,mBAAy,UAAZ,eAAyB,CAA3D,C;UAA8D,gBAAS,QAAT,C;QAC9D,iB;;QAEA,YAAy,QAAS,kBAAS,SAAT,C;QAErB,mBAAiB,4BAAU,K;QAC3B,IAAI,sDAA+B,kDAAnC,C;UAAgE,gBAAS,QAAT,C;QACrD,8BAAAX,YAAW,C;;MAVf,qB;K;0CAcJ,oB;MACI,MAAM,6BAAsB,iDAA+C,cAA/C,cAA8D,UAAAL,SAAK,CAA9D,wBAA2F,QAA3F,MAAtB,C;K;;ICpKd,yC;MACI,iBAAiB,QAAS,mB;MAC1B,IA2DA,OA3DI,MA2DH,8BAAO,CAAP,EAAD,kCA3DA,C;QACI,OAAO,wBAAwB,MAAxB,EAAGC,QAAhC,EAA0C,UAA1C,C;;MAEX,IAwDA,OAxDI,UAWDH,8BAAO,CAAP,EAAD,kCAxDA,C;QACI,OAAO,sBAAsB,MAAtB,EAA8B,QAA9B,C;;MAGX,aAAa,WAAS,UAAT,C;MACb,IAAM,WAAW,MAAX,CAAD,KAAyB,eAAe,MAAf,CAAzB,CAAD,cAAoD,CAAxD,C;QACI,OAAW,oBAAS,CAAb,sD;;MAEX,OAAO,M;K;IAGX,+D;MACI,IAAI,QAAS,aAAT,IAA0B,WAAW,UAAx,eAAwB,CAAtD,C;QAA0D,MAAM,gCAAYB,uCAAzB,C;MACHE,OAAO,M;K;IAGX,iD;MACI,WAAW,qBAAW,CAAX,C;MACX,IA5CA,OAAtCI,IAAK,mBAsCR,8BAAO,CAAP,EAAD,kCAtCA,C;QAEI,OAA8D,uBAAtD,oBAAS,QAAS,yDAAoC,C;;QAE9D,OAAO,cAAc,cAAc,MAAd,EAA5B,IAAtB,CAAd,EAA2C,sBAAW,IAAX,CAA3C,C;;K;IAIf,2C;MACI,IA6BA,OA7BI,QA6BH,8BAAO,CAAP,EAAD,kCA7BA,C;QACI,OAAkB,aAAT,QAAS,kCAAX,a;;MAEX,OAAO,qBAAqB,OAARb,EAA8B,QAA9B,C;K;IAGX,qD;MACI,IA5BA,OAAtBI,SA5BH,8BAAO,CAAP,EAAD,kCAtBA,C;QACI,IAAI,kBAAa,SAAb,CAAJ,C;UAA4B,OAAO,gCAAS,K;QAC5C,OAAmB,aAAV,SAAU,kCAAZ,a;;MAEX,IAkBA,OAIBI,SAkBH,8BAAO,CAAP,EAAD,kCAIBA,C;QACI,OAAiB,aAAV,SAAU,kC;;MAErB,OAAO,qBAAqB,SAARb,EAAGC,SAAhC,C;K;IAGX,kD;MACI,aAAa,kBAAW,QAAX,C;MACb,IAAK,WAAW,QAAX,CAAD,KAA2B,WAAW,QAAX,CAAqB,MAAhD,eAAwD,CAA5D,C;QACI,eAAe,iCAAW,OAAx,YAA6B,iCAAW,OAAx,EAA7B,C;QACf,eAAe,oCAAW,OAAx,YAA6B,oCAAW,OAAx,EAA7B,C;QACf,OjN8C4C,aiN9CrC,QjN8CqC,4BiN9CrC,ajN9QoC,aiNRZ,QjNQY,2BiNRpC,C;;MAEX,OjNM+C,aiNNxC,MjNMwC,2B;K;IiNHnD,gC;MAEI,cAAC,uCAAO,CAAP,EAAD,kC;K;qFChEJ,yB;MAAA,yC;MAAA,wB;QA4CI,WAAW,8B;QAjC6B,KakCx,E;QAICA,OAmCO,IAAK,a;O;KA9ChB,C;uFAeA,4B;MAYI,WAAW,mB;MACX,O;MACA,OAAO,IAAK,a;K;uFAGhB,4B;MAYI,WAAW,mB;MACX,O;MACA,OAAO,IAAK,a;K;IAYe,qC;MAAC,kB;MAAc,wB;K;;sCAR9C,Y;MAQgC,iB;K;sCARhC,Y;MAQ8C,oB;K;wCAR9C,2B;MAAA,sBAQgC,qCARhC,EAQ8C,8CAR9C,C;K;oCAAA,Y;MAAA,OAQgC,iDARhC,IAQ8C,8CAR9C,O;K;oCAAA,Y;MAAA,c;MAQgC,sD;MAAc,yD;MAR9C,a;K;kCAAA,iB;MAAA,4IAQgC,sCARhC,IAQ8C,4CAR9C,I;K;iGAUA,yB;MAAA,yC;MAkCA,8C;MAICA,wB;QA+CI,WAAW,8B;QACX,aAnC8C,KAmCjC,E;QAnCb,OAoCO,oBAAW,MAAX,EAAmB,IAAK,aAAxB,C;O;KAjDX,C;mGAgBA,yB;MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAmB,IAAK,aAAxB,C;O;KafX,C;mGakBA,yB;MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAmB,IAAK,aAAxB,C;O;KafX,C;Ipk/CA,2E;MASI,sC;MAAA,4C;K;IATJ,mGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,4DAaQ,kB;MACI,wBAAW,MAAX,C;K;IAIdZ,wF;IqKewC,sC;MACpC,0B;K;;IAGJ,kC;MAUI,OAA2C,CAA3C,2BAA6B,uBAA7B,EAAoC,KAApC,CAA2C,e;K;IAE/C,8B;K;kDAuBI,4B;MASI,MAAM,qCAA8B,8CAA9B,C;K;;IAW4B,8C;MAGtC,6B;MAEmD,UAMX,M;MAPxC,kBACmD,mE;MAEnD,eAC0B,K;MAE1B,cACwC,kE;MAExC,gBACmC,gB;K;iGAG/B,Y;MAAQ,0C;K;0DAEZ,kB;MACI,cAAy,I;MACZ,gBAAC,M;K;IAGsE,iG;MAAA,uB;QAExE,Q;QAAZ,qCAAY,8D;QACZ,sC



AAa,a;QAFb,OAGA,yB;O;K;2DAJJ,+B;MAAkD,OAAc,wDAATc,c;K;IAOyE,uH;MAAA,uB;QAEExG,Q;QAAf,i  
BA Ae,8F;QACf,eAAK,2B;QAA6B,mC;Q3M/FtB,gBAAT,Q;Q2MoG0D,kB;QAJzD,sBAAsB,SAAK,W;QAC3B,I  
AAI,eAAa,eAAjB,C;UAEL,iC;UACA,mBAAY,oCAAwB,eAAxB,EAAYc,kEAAzC,C;;UAGZ,mBAAY,kE;;QAEh  
B,oBAAa,e;QAZjB,OAcA,yB;O;K;6DAfJ,0C;MAAqF,OAAc,qEAAc,c;K;IAqBzB,mI;MAAA,qB;QACxD,yCA  
AgB,uB;QAGhB,qCAAY,Y;QACZ,uCAAc,E;QACIB,W;O;K;iEATA,iC;MAGwB,wCAAA,mCAAAb,EAAoC,kFA  
ApC,C;K;mDAQxB,Y;MAMuB,UADC,MACD,EAIH,MAJG,EAaK,M;MAjBxB,OAAO,IAAP,C;QAEI,aAAa,IA  
AK,S;QACF,SAAL,IAAK,O;QAAL,mB;UACyB,gBAArB,0D;UtKtBhB,U;UADP,yB;UsKuBe,OtKtBR,sF;;QsKq  
BC,WAAW,M;QAGX,IAAI,mDAAoB,MAApB,QAAJ,C;;YAIiB,SAAT,epKtJV,CoKsJuD,IpKtJvD,EoKsJ6D,Yp  
KtJ7D,EoKsJoE,IpKtJpE,EAA8C,KAA9C,C;;YoKuJQ,gC;cACE,IrKvJhB,oBDgDQ,WAAO,csKuG0B,CtKvG1B,  
CAAP,CChDR,C;cqKwJgB,Q;;cALI,O;;UAAAR,c;UAQA,IAAI,MAAM,yBAAV,C;YACI,IrKrKhB,oBDgDQ,WsK  
qHoB,0EtKrHpB,CChDR,C;;UqKwKY,gBAAc,gB;UACd,IAAK,oBAAW,MAAX,C;;K;;0ECxMrB,4B;MAyLI,Q  
ApLK,SAoLG,GApLoB,KAoLpB,I;MACR,IAAI,CArLC,SAqLD,GArlwB,KaqlxB,IAAiB,CAAjB,IAAsB,eArL  
E,KaqlF,MArLrB,SAqLL,C;QAA6C,a;;MARL7C,OAsLO,C;K;kEApLX,yB;MAAA,0B;MAAA,mC;QAgMI,QAv  
LK,SAuLG,GAvLe,KAuLf,I;QAvLR,OAAgC,OAwLzB,KAxLgB,KAwLX,GAAW,CAAC,CAAC,IAxLF,KAwLC  
,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAxLyB,C;O;KATpC,C;4EAWA,4B;MA  
uKI,QAIKK,SAKKG,GAlKoB,KakKpB,I;MACR,IAAI,CAnKC,SAmKD,GAnKwB,KAmKxB,IAAiB,CAAjB,IA  
AsB,eAnKE,KAmKF,MAAnKrB,SAmKL,C;QAA6C,a;;MAAnK7C,OAoKO,C;K;kEAIKX,yB;MAAA,4B;MAAA,m  
C;QA8KI,QArKK,SAqKG,GArKe,KaqKf,I;QArKR,OAAgC,QAsKzB,KAtKgB,KAsKX,GAAW,CAAC,CAAC,I  
AtKF,KAsKC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAtKyB,C;O;KATpC,C;4E  
AWA,4B;MAqJI,QAhJK,SAgJG,GAhJc,KagJd,I;MACR,IAAI,CAjJC,SAiJD,GAjJkB,KAiJlB,IAAiB,CAAjB,IAA  
sB,eAjJJ,KAiJI,MAjJrB,SAiJL,C;QAA6C,a;;MAjJ7C,OakJO,C;K;kEAhJX,4B;MA4JI,QAnJK,SAmJG,GAnJS,K  
AmJT,I;MAAnJR,OAoJO,KApJU,KAoJL,GAAW,CAAC,CAAC,IApJR,KAoJO,KAAmB,KAAK,CAAC,CAAD,IA  
AL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAIJX,yB;MA6NA,0B;MA7NA,mC;QAKkB,kBAAT,oBAAL,SAAK,  
C;QA6NL,QAAQ,gBA7Ne,Ka6Nf,C;QACR,IAAI,gBA9NmB,Ka8NnB,eAAiB,CAAjB,IAAsB,mBA9NH,Ka8N  
G,GAAa,WAAb,CAA1B,C;UAA6C,W;;QA9N7C,OA+NO,C;O;KApOX,C;kEAOA,4B;MAyOI,QAhOK,oBAAL,  
SAAK,CAgOG,QAhOU,KAgOV,C;MAhOR,OAiOO,MAjOW,KAiON,KAAa,MAjOP,KAiOO,CAAD,KAAmB,K  
AAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;4EA/NX,4B;MAiHI,QA5GK,SA4GG,G  
A5GoB,Ka4GpB,I;MACR,IAAI,CA7GC,SA6GD,GA7GwB,Ka6GxB,IAAiB,CAAjB,IAAsB,eA7GE,Ka6GF,MA  
7GrB,SA6GL,C;QAA6C,a;;MA7G7C,OA8GO,C;K;kEA5GX,yB;MAAA,0B;MAAA,mC;QAwHI,QA/GK,SA+GG  
,GA/Ge,Ka+Gf,I;QA/GR,OAAgC,OAgHzB,KAhHgB,KAgHX,GAAW,CAAC,CAAC,IAhHF,KAgHC,KAAmB,K  
AAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAhHyB,C;O;KATpC,C;4EAWA,4B;MA+FI,QA1F  
K,SA0FG,GA1FoB,Ka0FpB,I;MACR,IAAI,CA3FC,SA2FD,GA3FwB,Ka2FxB,IAAiB,CAAjB,IAAsB,eA3FE,K  
A2FF,MA3FrB,SA2FL,C;QAA6C,a;;MA3F7C,OA4FO,C;K;kEA1FX,yB;MAAA,4B;MAAA,mC;QAsGI,QA7FK,  
SA6FG,GA7Fe,Ka6Ff,I;QA7FR,OAAgC,QA8FzB,Ka9FgB,Ka8FX,GAAW,CAAC,CAAC,IA9FF,Ka8FC,KAA  
mB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,Ka9FyB,C;O;KATpC,C;4EAWA,4B;MA6EI,Q  
AxEK,SAwEG,GAxEc,KAwEd,I;MACR,IAAI,CAzEC,SAyED,GAzEkB,KAyEIB,IAAiB,CAAjB,IAAsB,eAzEJ,K  
AyEI,MAzErB,SAyEL,C;QAA6C,a;;MAzE7C,OA0EO,C;K;kEAxEX,4B;MAoFI,QA3EK,SA2EG,GA3ES,Ka2ET  
,I;MA3ER,OA4EO,Ka5EU,Ka4EL,GAAW,CAAC,CAAC,IA5ER,Ka4EO,KAAmB,KAAK,CAAC,CAAD,IAAL,  
CAAnB,CAAD,KAAkC,EAAID,K;K;4EA1EX,yB;MAqJA,0B;MARJA,mC;QAKkB,kBAAT,oBAAL,SAAK,C;QA  
qJL,QAAQ,gBArJe,KaqJf,C;QACR,IAAI,gBAtJmB,KAsJnB,eAAiB,CAAjB,IAAsB,mBAtJH,KAsJG,GAAa,WA  
Ab,CAA1B,C;UAA6C,W;;QAtJ7C,OAuJO,C;O;KA5JX,C;kEAOA,4B;MAiKI,QAxJK,oBAAL,SAAK,CAwJG,Q  
AxJU,KAwJV,C;MAxJR,OAyJO,MAzJW,KAyJN,KAAa,MAzJP,KAyJO,CAAD,KAAmB,KAAAM,CAAD,aAAL,  
CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;2EA vJX,4B;MAyCI,QApCA,SAoCQ,GApCY,KaOCZ,I;MA  
CR,IAAI,CArCJ,SAqCI,GArCgB,KaqChB,IAAiB,CAAjB,IAAsB,eArCN,KaqCM,MArC1B,SAqCA,C;QAA6C,a  
;;MARc7C,OAsCO,C;K;iEApCX,yB;MAAA,0B;MAAA,mC;QAgDI,QAvCA,SAuCQ,GAvCO,KAuCP,I;QAvCR,  
OAAwB,OAwCjB,KAxCQ,KAwCH,GAAW,CAAC,CAAC,IAxCV,KAwCS,KAAmB,KAAK,CAAC,CAAD,IAA  
L,CAAnB,CAAD,KAAkC,EAAID,KAxCiB,C;O;KAT5B,C;4EAWA,4B;MAuBI,QAlBA,SAkBQ,GAIBY,KakBZ,  
I;MACR,IAAI,CAnBJ,SAmBI,GAnBgB,KAmBhB,IAAiB,CAAjB,IAAsB,eAnBN,KAmBM,MAAnB1B,SAmBA,C;

QAA6C,a;;MAnB7C,OAoBO,C;K;mEAIBX,yB;MAAA,4B;MAAA,mC;QA8BI,QArBA,SAqBQ,GArBO,KAqBP,I  
;QArBR,OAAwB,QAsBjB,KAtBQ,KAsBH,GAAW,CAAC,CAAC,IAtBV,KAsBS,KAAmB,KAAK,CAAC,CAAD,  
IAAL,CAAnB,CAAD,KAAkC,EAAID,KAtBiB,C;O;KAT5B,C;4EAWA,4B;MAKI,QAAQ,YAAO,KAAP,I;MAC  
R,IAAI,aAAS,KAAT,IAAiB,CAAjB,IAAsB,eAAI,KAAJ,MAAa,SAAvC,C;QAA6C,a;;MAC7C,OAAO,C;K;mEA  
GX,4B;MASI,QAAQ,YAAO,KAAP,I;MACR,OAAO,KAAK,QAAW,CAAC,CAAC,IAAM,KAAP,KAAmB,KAA  
K,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAGX,yB;MAwEA,0B;MAxEA,mC;QAKkB,kBA  
AT,oBAAL,SAAK,C;QAwEL,QAAQ,gBAxEe,KAwEf,C;QACR,IAAI,gBAzEmB,KAyEnB,eAAiB,CAAjB,IAAs  
B,mBAzEH,KAyEG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAzE7C,OA0EO,C;O;KA/EX,C;kEAOA,4B;MAoFI,Q  
A3EK,oBAAL,SAAK,CA2EG,QA3EU,KA2EV,C;MA3ER,OA4EO,MA5EW,KA4EN,KAAa,MA5EP,KA4EO,CA  
AD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;6EA1EX,yB;MasDA,0B;  
MatDA,mC;QAKS,cAAe,oBAAN,KAAM,C;QAsDpB,QAtDA,SAsDQ,KAAO,OAAP,C;QACR,IAvDA,SAuDI,K  
AAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAAJ,GAvD1B,SAuD0B,CAA1B,C;UAA6C,W;;QAvD7C,OAwDO,  
C;O;KA7DX,C;mEAOA,yB;MAAA,0B;MAAA,mC;QASS,cAAU,oBAAN,KAAM,C;QAYDf,QAzDA,SAyDQ,Q  
AAO,OAAP,C;QAzDR,OAAYB,OA0DIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,  
CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CA1DkB,S;O;KAT7B,C;6EAWA,yB;MAoCA,0B;MApCA,mC;Q  
AKS,cAAe,oBAAN,KAAM,C;QAoCpB,QApCA,SAoCQ,KAAO,OAAP,C;QACR,IARCA,SAqCI,KAAS,OAAT,e  
AAiB,CAAjB,IAAsB,mBAAI,OAAJ,GArC1B,SAqC0B,CAA1B,C;UAA6C,W;;QArC7C,OAsCO,C;O;KA3CX,C;  
mEAOA,yB;MAAA,4B;MAAA,mC;QASS,cAAU,oBAAN,KAAM,C;QAuCF,QAvCA,SAuCQ,QAAO,OAAP,C;Q  
AvCR,OAAYB,QAwCIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,Y  
AAkC,EAAIC,CAAX,CAAL,CAXCkB,S;O;KAT7B,C;6EAWA,yB;MAkBA,0B;MA1BA,mC;QAKS,cAAe,oBAA  
N,KAAM,C;QAKpB,QAlBA,SAkBQ,KAAO,OAAP,C;QACR,IAnBA,SAmBI,KAAS,OAAT,eAAiB,CAAjB,IAA  
sB,mBAAI,OAAJ,GAnB1B,SAmB0B,CAA1B,C;UAA6C,W;;QAnB7C,OAoBO,C;O;KazBX,C;mEAOA,4B;MAS  
S,cAAU,oBAAN,KAAM,C;MAqBf,QArBA,SAqBQ,QAAO,OAAP,C;MArBR,OAsBO,MAAK,YAAa,MAAM,O  
AAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAtBkB,Q;K;6EAE7  
B,yB;MAAA,0B;MAAA,mC;QAKI,QAAQ,cAAO,KAAP,C;QACR,IAAI,cAAS,KAAT,eAAiB,CAAjB,IAAsB,m  
BAAI,KAAJ,GAAa,SAAb,CAA1B,C;UAA6C,W;;QAC7C,OAAO,C;O;KAPX,C;mEUA,4B;MASI,QAAQ,iBAA  
O,KAAP,C;MACR,OAAO,MAAK,UAAa,MAAM,KAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,  
YAAkC,EAAIC,CAAX,CAAL,C;K;kEAGX,yB;M5GmqB2C,0B;M4GnqB3C,mC;QAWI,QAAQ,YAAO,K;QACJ,  
iBAAS,G;QAAT,S;UAAAsB,O5GupB0C,W4GvpB1C,C5GupB0C,C4GvpB1C,K5GupB0C,W4GvpBhC,K5GupBg  
C,C;;Q4GvpB3E,OAAO,OAAGD,IAAI,KAAPD,GAA+D,C;O;KAZ1E,C;mEAeA,yB;M5G0H6C,0B;M4G1H7C,m  
C;QAqCI,QA1BK,SA0BG,GA1BY,K;QA2BT,iBAAK,G;QAAL,S;UAAAY,O5GoF0B,W4GpF1B,C5GoF0B,C4Gp  
F1B,K5GoF0B,W4G/G7B,K5G+G6B,C;;Q4G/GjD,OA2BO,OAAsC,IA3BzB,KA2Bb,GAAqD,C;O;KAtChE,C;mE  
AaA,yB;M5G6G6C,0B;M4G7G7C,mC;QAwBI,QAbA,SAaQ,GAbO,K;QAcJ,iBAAK,G;QAAL,S;UAAAY,O5GoF0  
B,W4GpF1B,C5GoF0B,C4GpF1B,K5GoF0B,W4GIGIC,K5GkGkC,C;;Q4GIGjD,OAcO,OAAsC,IAd9B,KAcR,GA  
AqD,C;O;KazBhE,C;mEAaA,yB;M5GgG6C,0B;M4GhG7C,mC;QAWI,QAAQ,YAAO,K;QACJ,iBAAK,G;QAAL  
,S;UAAAY,O5GoF0B,W4GpF1B,C5GoF0B,C4GpF1B,K5GoF0B,W4GpFhB,K5GoFgB,C;;Q4GpFjD,OAAO,OAAs  
C,IAAI,KAA1C,GAAqD,C;O;KAZhE,C;4ECtVA,yB;MAAA,8B;MAAA,4B;QAOyC,Q;QAAA,gFAAoB,C;O;KA  
P7D,C;ICM0B,4C;MA+CtB,qC;MA/CuB,kB;MAAgB,kB;MAAgB,kB;MAMvD,iBAAsB,iBAAU,UAAV,EAAiB,  
UAAjB,EAAwB,UAAxB,C;K;0CAEtB,+B;MjNWA,IAAI,EiNViB,CAAT,sBAAY,GAAZ,KAA4C,CAAT,sBAAY  
,GAA/C,MAA+E,CAAT,sBAAY,GAAIF,CjNUR,CAAJ,C;QACI,ciNVI,2E;QjNWJ,MAAM,gCAAYB,OAAQ,WA  
AjC,C;;MiNTN,OAAO,CAAA,KAAM,IAAI,EAAV,KAAGB,KAAM,IAAI,CAA1B,IAA+B,KAA/B,I;K;uCAGX,  
Y;MAGkC,OAAE,UAAF,oBAAS,UAAT,SAAGB,U;K;qCAEID,iB;MAEWB,gB;MADpB,IAAI,SAAS,KAAb,C;Q  
AAoB,OAAO,I;MACP,iE;MAAD,mB;QAA6B,OAAO,K;;MAAvD,mBAAmB,M;MACnB,OAAO,IAAK,UAAAL,K  
AAgB,YAAa,U;K;uCAGxC,Y;MAA+B,qB;K;8CAE/B,iB;MAAoD,wBAAU,KAAM,UAAhB,I;K;gDAEpD,wB;M  
AKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAAd,IACf,IAAK,MAAL,IAAc,KADtB,  
C;K;gDAGJ,+B;MAKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAAd,KACd,IAAK,M  
AAL,GAAa,KAAb,KAAsB,IAAK,MAAL,KAAc,KAAAd,IACf,IAAK,MAAL,IAAc,KADrB,CADc,CAAvB,C;K;IA  
IJ,mC;MAAA,uC;MACI,2BAIuC,G;MAEvC,eAIoC,uCAA0B,M;K;;;IAXIE,+C;MAAA,8C;QAAA,6B;;MAAA,uC

;K;;IA9CA,iD;MAAA,uD;MAG6C,0BAAK,KAAL,EAAy,KAAZ,EAAmB,CAAnB,C;MAH7C,Y;K;IA6DJ,qC;M  
AAA,yC;K;8CAEL,Y;MAC2B,yBAAC,CAAd,EAAiB,CAAjB,EAAoB,EAAPB,C;K;;IAH/B,iD;MAAA,gD;QAAA  
,+B;;MAAA,yC;K;4FCxDI,yB;MAAA,2D;MAAA,4B;QAAQ,MAAM,6BAAoB,6BAApB,C;O;KAAAd,C;;;ICSJ,u  
B;MAG2C,+BAAoB,KAApB,C;K;4EAE3C,wC;MAO4F,sB;K;IAE5F,6C;MAAA,e;MAAA,iB;MAAA,uB;K;IAA  
A,2C;MAAA,8C;O;MAKI,wF;MAKA,sF;MAMA,wE;K;;IAXA,yD;MAAA,iC;MAAA,iD;K;;IAKA,wD;MAAA,i  
C;MAAA,gD;K;;IAMA,iD;MAAA,iC;MAAA,yC;K;;IAhBJ,uC;MAAA,iJ;K;;IAAA,4C;MAAA,a;MAAA,c;UAAA,  
sD;aAAA,a;UAAA,qD;aAAA,M;UAAA,8C;;UAAA,gE;;K;;IAyBA,+B;MAAA,mC;K;;IAAA,2C;MAAA,0C;QA  
AA,yB;;MAAA,mC;K;IAGoC,qC;MACHC,qBAAsC,W;MACtC,gBAA2B,iC;K;uFAGvB,Y;MAMW,Q;MALP,IA  
Al,kBAAW,iCAAF,C;QACI,gBAAS,mC;QACT,qBAAC,I;;MAGIB,OAAO,gF;K;6CAGf,Y;MAAwC,yBAAW,iC;  
K;wCAEnD,Y;MAAkC,OAAI,oBAAJ,GAA2B,SAAN,UAAM,CAA3B,GAA2C,iC;K;8CAE7E,Y;MAAkC,+BAA  
oB,UAApB,C;K;;IAGG,oC;MAAC,4B;K;wEAAA,Y;MAAA,2B;K;kDAEtC,Y;MAAwC,W;K;6CAExC,Y;MAAk  
C,OAAAM,SAAN,UAAM,C;K;;oFC2C5C,yB;MAAA,gD;MAAA,4B;QAM6C,OAAmB,aAAlB,YAAy,GAAM,C;  
O;KANhE,C;oGAQA,yB;MhH7FA,4B;MgH6FA,4B;QAMqD,OhH7FM,YgH6FL,YAAy,GhH7FP,CgH6FN,GAA  
6C,EAA7C,I;O;KANrD,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAmB,sBAAIB,YAAW,GAAO,C;O;KA  
NzE,C;8FAQA,yB;MAAA,0D;MAAA,0B;MAAA,4B;QAOmD,OAAuC,OAApB,kBAAIB,YAAy,GAAM,CAAo  
B,C;O;KAP1F,C;4FASA,yB;MAAA,wD;MAAA,0B;MAAA,4B;QAOkD,OAA2B,OAAAnB,iBAAR,SAAQ,CAAm  
B,C;O;KAP7E,C;IAUA,2C;MAaI,OAA+E,OAA9E,SAAQ,KAAI,WAAa,CAAjB,CAAR,GAakD,CAAIB,YAAy,  
GAAM,MAAK,CAAL,IAAU,WAAa,CAAvB,CAA4B,C;K;IAEnF,4C;MAaI,OAA+E,OAA9E,SAAQ,IAAI,CAAJ,  
IAAS,WAAa,CAAtB,CAAR,GAAwD,CAAIB,YAAy,GAAM,OAAK,WAAa,CAAIB,CAAsB,C;K;oFAEnF,yB;M  
AAA,gD;MAAA,4B;QAM8C,OAAqB,aAApB,YAAy,KAAQ,C;O;KANnE,C;oGAQA,yB;MhHtKA,4B;MgHsKA  
,4B;QAOI,OhHvKuD,YgHuKtD,YAAy,KhHvK0C,CgHuKvD,GAA+C,EAA/C,I;O;KAPJ,C;sGASA,yB;MAAA,k  
E;MAAA,4B;QAMuD,OAAqB,sBAApB,YAAW,KAAS,C;O;KAN5E,C;8FAQA,yB;MAAA,0D;MAAA,4B;MAA  
A,4B;QAOqD,OAAyC,QAAPB,kBAApB,YAAy,KAAQ,CAAoB,C;O;KAP9F,C;4FASA,yB;MAAA,wD;MAAA,  
4B;MAAA,4B;QAOoD,OAA2B,QAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP/E,C;IAUA,2C;MAaI,OAAoF,QAAnF  
,SAAQ,KAAI,WAAa,EAAjB,CAAR,GAAqD,CAAPB,YAAy,KAAQ,MAAK,EAAL,IAAW,WAAa,EAAxB,CAA  
8B,C;K;IAExF,4C;MAaI,OAAoF,QAAnF,SAAQ,IAAL,EAAJ,IAAU,WAAa,EAAvB,CAAR,GAA4D,CAAPB,YA  
AY,KAAQ,OAAK,WAAa,EAAIB,CAAuB,C;K;0EpNIRxY,yB;MAaA,kF;MAbA,wB;QAUbI,IAAI,CAbI,KAAR,C;  
UACI,cAda,qB;UAeb,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAZbD,C;0EAaA,yB;MAAA,kF;MAAA,qC;QAU  
IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAZd,C;sFAgBA,yB;MAW  
A,kF;MAXA,wB;QAQW,yB;QAEp,IAfsB,KAeIB,QAAJ,C;UACI,cAhB2B,0B;UAIb3B,MAAM,8BAAyB,OAAQ,  
WAAjC,C;;UAEN,wBAnBkB,K;;QAAtB,4B;O;KARJ,C;wFAWA,yB;MAAA,kF;MAAA,qC;QAYI,IAAI,aAAJ,C;  
UACI,cAAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;UAEN,OAAO,K;;O;KAhBf,C;oEAoBA,yB;MAaA,4E;  
MAbA,wB;QAUbI,IAAI,CAbE,KAaN,C;UACI,cAdW,e;UAeX,MAAM,2BAAsB,OAAQ,WAA9B,C;;O;KAZbD,C  
;sEAaA,yB;MAAA,4E;MAAA,qC;QAU,IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,W  
AA9B,C;;O;KAZd,C;kFAgBA,yB;MAcA,4E;MAdA,wB;QAWW,uB;QAEp,IAfoB,KAehB,QAAJ,C;UACI,cAhBy  
B,0B;UAIbZB,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;QAAPB,0B;O;KAXJ,C;oFAcA,yB;MA  
AA,4E;MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,OAAO  
,K;;O;KAhBf,C;oEAqBA,yB;MAAA,4E;MAAA,0B;QAMiD,MAAM,2BAAsB,OAAQ,WAA9B,C;O;KANvD,C;I  
wCnHiC,uB;MA2D7B,8B;MA1DA,kB;K;mFAS8B,Y;MAAQ,iD;K;mFAMR,Y;MAAQ,gD;K;wFAItC,yB;MAAA,  
gB;MAAA,8B;MAAA,mB;QAWgB,Q;QADR,mB;UADJ,OACiB,I;;UADjB,OAEY,2E;O;KAXhB,C;uCAcA,Y;M  
AQQ,kBADE,UACF,kB;QADJ,OACkB,UAAM,U;;QADxB,OAEY,I;K;gCAGhB,Y;MAOQ,kBADE,UACF,kB;Q  
ADJ,OACkB,UAAM,W;;QADxB,OAEY,sBAAU,UAAV,O;K;IAKhB,4B;MAAA,gC;K;wHAKI,yB;MAAA,iC;M  
AAA,wB;QAOI,uBAAO,KAAp,C;O;KAPJ,C;wHASA,yB;MAAA,kD;MAAA,iC;MAAA,4B;QAOI,uBAAO,cAA  
c,SAAd,CAAP,C;O;KAPJ,C;;IADJ,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;IAwBsB,mC;MACIB,0B;K;sCAGA,i  
B;MAA4C,+CAAoB,uBAAa,KAAM,UAAnB,C;K;wCACHE,Y;MAA+B,OAAU,SAAV,cAAU,C;K;wCACzC,Y;  
MAAkC,oBAAU,cAAV,M;K;;gCAF1C,Y;MAAA,c;MAOI,sD;MAPJ,a;K;8BAAA,iB;MAAA,2IAOI,sCAPJ,G;  
K;IAmGA,kC;MAOI,OAAO,mBAAQ,SAAR,C;K;IAEX,mC;MAQI,IAAI,8CAAJ,C;QAA6B,MAAM,eAAM,U;K;  
gFAG7C,yB;MAAA,4B;MAAA,qB;MAxCQ,kD;MAwCR,wB;QAOW,Q;;UACI,OAIDH,WAKDW,OAIDX,C;;UA

mDN,gC;YACS,OA3CH,WAAO,cA2CI,CA3CJ,CAAP,C;;YAwCD,O;;QAAP,W;O;KAPJ,C;kFAcA,yB;MAAA,4B;MAAA,qB;MAtdQ,kD;MASDR,mC;QAOW,Q;;UACI,OAHEH,WAgEW,gBAhEX,C;;UAiEN,gC;YACS,OAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;YAsDD,O;;QAAP,W;O;KAPJ,C;8EAgBA,yB;MAAA,oD;MAAA,gB;MAAA,8B;MAAA,4B;QAUW,Q;QADP,yB;QACA,OAAO,gF;O;KAVX,C;+EAaA,yB;MAAA,gB;MAAA,8B;MAAA,uC;QAegB,UADL,M;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,yF;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAdJ,C;kFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,0C;QAUW,Q;QADP,IAAL,mBAAJ,C;UAAe,OAAO,Y;QAcTb,OAAO,gF;O;KAVX,C;qEAaA,yB;MAAA,gB;MAAA,8B;MAAA,kD;QAIb0B,UAdf,M;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,mBAAU,gFAAV,C;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAhBJ,C;mEAwBA,yB;MAAA,4B;MAAA,gB;MAAA,8B;MAAA,uC;YAe8C,I;YADnC,M;QACH,wB;UAAa,gB;UAAO,SA7JhB,WA6JwB,UAAU,gFAAV,CA7JxB,C;;UA8JI,oBAAO,eAAP,C;QAFZ,a;O;KAdJ,C;gFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,iC;MA1GA,qB;MAtdQ,kD;MAgKR,uC;QAWW,Q;QACH,wB;UA/GG,U;;YA+GkC,U;YA9G9B,SAhEH,gBA8KuB,UAAU,sFAAV,CA9KvB,C;;YAiEN,gC;cACS,SAzDH,gBAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UA+GU,a;;UACL,uBAAO,eAAP,C;QAFZ,W;O;KAXJ,C;wEAiBA,yB;MAAA,4B;MAAA,uC;QAcW,Q;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,gB;;UACO,OAnMX,WAmMmB,UAAU,SAAV,CAnMnB,C;;QAIrMR,W;O;KAdJ,C;wFAoBA,yB;MA/IA,4B;MAAA,qB;MAtdQ,kD;MAqMR,uC;QAWW,Q;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,gB;;UApJL,U;;YACI,SAhEH,WAOkNB,oBApNIB,C;;YAiEN,gC;cACS,SAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UAqJK,a;;QAFZ,W;O;KAXJ,C;4EAmBA,6B;MAUI,Q;MAAA,iD;QAAYB,Y;;MACzB,OAAO,S;K;4EAGX,yB;MAAA,gB;MAAA,8B;MAAA,oC;QAU0B,Q;QAAtB,IAAL,mBAAJ,C;UAAe,OAAO,gFAAP,C;;QACf,OAAO,S;O;KAXX,C;IrCtTgC,sC;MAAC,uB;QAAA,UAAkB,kC;mBAA4C,O;;K;;0DAE/F,yB;MAAA,2D;MAAA,mB;QAKoC,MAAM,8B;O;KAL1C,C;oEAOA,yB;MAAA,2D;MAAA,yB;QAMkD,MAAM,6BAAoB,sCAAmC,MAAvD,C;O;KANxD,C;gEAUA,iB;MAUI,OAAO,O;K;kEAGX,4B;MAUI,OAAO,gB;K;oEAGX,2B;MAUI,OAAgB,MAAT,QAAS,C;K;oEAGpB,4B;MAUI,gB;MACA,OAAO,S;K;kEAGX,4B;MAWI,MAAM,SAAN,C;MACA,OAAO,S;K;kEAGX,4B;MAUI,OAAO,MAAM,SAAN,C;K;sEAGX,gC;MAWI,OAAW,UAAU,SAAV,CAAJ,GAAqB,SAArB,GAA+B,I;K;8EAG1C,gC;MAWI,OAAW,CAAC,UAAU,SAAV,CAAL,GAAsB,SAAtB,GAAgC,I;K;wEAG3C,yB;MAWI,iBAc,CAAd,UAAsB,KAAtB,U;QACI,OAAO,KAAP,C;;K;wEkNjJR,iB;MAGkF,Y;K;ICa9C,6B;MACHC,kB;MACA,oB;K;8BAGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,M;K;;gCAvB7C,Y;MAGBI,iB;K;gCAhBJ,Y;MAiBI,kB;K;kCAjBJ,yB;MAAA,gBAGBI,qCAhBJ,EAiBI,wCAjBJ,C;K;8BAAA,Y;MAAA,c;MAGBI,sD;MACA,uD;MAjBJ,a;K;4BAAA,iB;MAAA,4IAgBI,sCAhBJ,IAiBI,wCAjBJ,I;K;IA0BA,6B;MAMoD,gBAAK,SAAL,EAAW,IAAX,C;K;IAEpD,8B;MAI8C,iBAAO,eAAP,EAAC,gBAAd,E;K;IAiBD,sC;MACzC,kB;MACA,oB;MACA,kB;K;gCAGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,UAAoB,UAApB,M;K;kCAxB7C,Y;MAGBI,iB;K;kCAhBJ,Y;MAiBI,kB;K;kCAjBJ,Y;MAkBI,iB;K;oCAIBJ,gC;MAAA,kBAGBI,qCAhBJ,EAiBI,wCAjBJ,EAkBI,qCAIBJ,C;K;gCAAA,Y;MAAA,c;MAGBI,sD;MACA,uD;MACA,sD;MAIBJ,a;K;8BAAA,iB;MAAA,4IAgBI,sCAhBJ,IAiBI,wCAjBJ,IAkBI,sCAIBJ,I;K;IA2BA,8B;MAImD,iBAAO,eAAP,EAAC,gBAAd,EAAsB,eAAtB,E;K;InOIE1B,qB;MAErB,6B;MAFkG,gB;K;IAEIG,2B;MAAA,+B;MACI,iBAGoC,UAAAM,CAAN,C;MAEpC,iBAGoC,UAAAM,MAAN,C;MAEpC,kBAGmC,C;MAEnC,iBAGkC,C;K;;;IANtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;kGAsBA,iB;MAOmE,OAaA,0BAqP1C,SAAL,GAAiB,GArP8B,EAAU,KAqPpD,KAAL,GAAiB,GArP8B,C;K;sGAehF,iB;MAM2D,OAAa,0BA6OIC,SAAL,GAAiB,GA7OsB,EAAU,KE8O5C,KAAL,GAAiB,KF9OsB,C;K;sGAExE,yB;MAoQA,6B;MCRQA,8C;MDCA,wB;QAMyD,OCAS,YAAiB,CDuQhD,cAAU,SAAL,GAAiB,GAAtB,CCvQgD,MAAjB,EDAe,KCAc,KAA7B,C;O;KDNIE,C;sGAQA,yB;MASQA,WAS6D,wB;MAT7D,+B;MkBvQA,gD;MIBCA,wB;QAM0D,OkBAS,aAAkB,CIByQhD,eAAW,oBAAL,SAAK,CAAL,UAAAN,CkBzQgD,MAAIB,EIBAgB,KkBAC,KAA9B,C;O;KIBNnE,C;4FAQA,yB;MAoPA,6B;MApPA,wB;QAEsD,OCMD,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCrP4B,MAAK,GAAW,CDqP5C,cA3PsC,KA2P5B,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KDRrD,C;4FAGA,yB;MAiPA,6B;MAjPA,wB;QAEuD,OCGF,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCrP4B,MAAK,GAAW,CCsP5C,cFzPuC,KEyP7B,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KDLrD,C;4FAGA,yB;MA8OA,6B;MA9OA,wB;QAEqD,OCAA,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCrP4B,MAAK,GDAI,KCAO,KAAx,IAAf,C;O;KDFrD,C;4FAGA,yB;MAqPA,WAS6D,wB;MAT7D,+B;MArPA,wB;QAEuD,OkBAA,eAAW,CIB4P7B,eAAW,oBAAL,SAAK,CAAL,UAAAN,CkB5P6B,MAAK,KIBAI,KkBAO,KAAx,CAAhB,C;O;KIBFvD,C;8FAIA,yB;MAuOA,6B;MAvOA,wB;QAEuD,OCMD,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6

B,MAAK,GAAY,CDwO9C,cA9OwC,KA8O9B,KAAL,GAAiB,GAAtB,CCxO8C,MAAZ,IAAf,C;O;KDRtD,C;8F  
AGA,yB;MAoOA,6B;MApOA,wB;QAEwD,OCGF,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6B,MA  
AK,GAAY,CCyO9C,cF5OyC,KE4O/B,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KDLtD,C;8FAGA,yB;  
MAiOA,6B;MAjOA,wB;QAEsD,OCAA,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6B,MAAK,GDAK,  
KCAO,KAAZ,IAAf,C;O;KDFtD,C;8FAGA,yB;MAwOA,WAS6D,wB;MAT7D,+B;MAxOA,wB;QAEwD,OkBAA  
,eAAW,CIB+O9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB/O8B,MAAK,UIBAK,KkBAO,KAAZ,CAAhB,C;O;  
KIBFxD,C;8FAIA,yB;MA0NA,6B;MA1NA,wB;QAEuD,OCMD,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,G  
AAtB,CC3N6B,MAAK,EAAY,CD2N9C,cAjOwC,KAiO9B,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;  
KDRtD,C;8FAGA,yB;MAuNA,6B;MAvNA,wB;QAEwD,OCGF,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,GA  
AtB,CC3N6B,MAAK,EAAY,CC4N9C,cF/NyC,KE+N/B,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KD  
LtD,C;8FAGA,yB;MAoNA,6B;MApNA,wB;QAEsD,OCAA,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,GAAtB  
,CC3N6B,MAAK,EDAK,KCAO,KAAZ,CAAf,C;O;KDFtD,C;8FAGA,yB;MA2NA,WAS6D,wB;MAT7D,+B;MA  
3NA,wB;QAEwD,OkBAA,eAAW,CIBkO9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBIO8B,MAAK,UIBAK,Kk  
BAO,KAAZ,CAAhB,C;O;KIBFxD,C;0FAIA,yB;MA6MA,6B;MCvMA,4C;MDNA,wB;QAEqD,OCMD,WD8MjB  
,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,ED8MjB,cApNoC,KAO1B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KDRp  
D,C;0FAGA,yB;MA0MA,6B;MCvMA,4C;MDHA,wB;QAEsD,OCGF,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,C  
C9MiB,EC+MjB,cFINqC,KEkN3B,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KDLpD,C;0FAGA,yB;MAuMA,6B;MC  
vMA,4C;MDAA,wB;QAEoD,OCAA,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EDakB,KCAIB,C;O;KDF  
pD,C;0FAGA,yB;MA8MA,WAS6D,wB;MAT7D,+B;MkB9MA,8C;MIBAA,wB;QAEsD,OkBAA,YIBqNjB,eAA  
W,oBAAL,SAAK,CAAL,UAAN,CkBrNiB,EIBAmB,KkBAnB,C;O;KIBFtD,C;0FAIA,yB;MAgMA,6B;MCILA,kD  
;MDdA,wB;QAMqD,OCcD,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDqLjB,cAnMoC,KAmM1B,KAAL,  
GAAiB,GAAtB,CCrLiB,C;O;KDPbPD,C;0FAOA,yB;MAyLA,6B;MCILA,kD;MDPA,wB;QAMsD,OCOF,cDqLjB  
,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,ECsLjB,cF7LqC,KE6L3B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KDbpD,C;  
0FAOA,yB;MAkLA,6B;MCILA,kD;MDAA,wB;QAMoD,OCAA,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,  
EDakB,KCAIB,C;O;KDNpD,C;0FAOA,yB;MAqLA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;MIBAA,wB;QAMsD  
,OkBAA,eIBwLjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBxLiB,EIBAmB,KkBAnB,C;O;KIBNtD,C;oGAQA,y  
B;MAmKA,6B;MCvMA,4C;MDoCA,wB;QAMiD,OCxCG,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,ED  
8MjB,cAtKqC,KAsK3B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KDKCpD,C;oGAOA,yB;MA4JA,6B;MCvMA,4C;  
MD2CA,wB;QAMkD,OC/CE,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EC+MjB,cFhKsC,KEgK5B,KAA  
L,GAAiB,KAAtB,CD/MiB,C;O;KDyCpD,C;oGAOA,yB;MAqJA,6B;MCvMA,4C;MDkDA,wB;QAMgD,OCtDI,  
WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EDsDmB,KCtDnB,C;O;KDgDpD,C;oGAOA,yB;MAwJA,WAS  
6D,wB;MAT7D,+B;MkBrLA,8C;MIBsDA,wB;QAMkD,OkB1DI,YIBqNjB,eAAW,oBAAL,SAAK,CAAL,UAAN  
,CkBrNiB,EIB0DoB,KkB1DpB,C;O;KIBoDtD,C;0FAQA,yB;MASIA,6B;MCILA,kD;MDiPJ,0B;MAAA,+B;MAr  
MI,wB;QAQ6C,OAwMR,eAAW,OCtPI,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDqLjB,cAv14B,KAUllB,  
KAAL,GAAiB,GAAtB,CCrLiB,CA4Lf,KD0DW,CAAX,C;O;KAhNrC,C;0FASA,yB;MA6HA,6B;MCILA,kD;MC  
kPJ,4B;MAAA,iC;MF7LI,wB;QAQ+C,OEgMR,gBAAY,QDvPC,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,E  
CsLjB,cF/H8B,KE+HpB,KAAL,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C;O;KFXMvC,C;0FASA,yB;MA  
oHA,6B;MCILA,kD;MD8DA,wB;QAQ2C,OChES,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDgES,KChET  
,C;O;KDwDpD,C;0FASA,yB;MAqHA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;MIBgEA,wB;QAQ6C,OkBIES,eIB  
wLjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBxLiB,EIBkEU,KkBIEV,C;O;KIB0DtD,C;0EAUA,yB;MAAA,0B;  
MAAA,+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,0B;MAAA,+  
B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,8C;MAiFA,6B;MAjF  
A,wB;QAE8D,0BAwF3B,cAAU,SAAL,GAAiB,GAAtB,CAXF2B,EAwF3B,cAXFoD,KAwF1C,KAAL,GAAiB,GA  
AtB,CAXF2B,C;O;KAF9D,C;wGAIA,yB;MAAA,yC;MA6EA,6B;MA7EA,wB;QAQiE,aA8E9B,cAAU,SAAL,GA  
AiB,GAAtB,CA9E8B,EA8E9B,cA9EkD,KA8ExC,KAAL,GAAiB,GAAtB,CA9E8B,C;O;KARjE,C;0FAUA,yB;M  
AAA,+B;M8LIPJ,0B;M9LkPI,wB;QAEEmD,sB8LjPgC,O9LiP1B,IAAK,K8LjPX,G9LiPoB,KAAM,K8LjPM,C9LiP  
hC,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;M8LhPJ,0B;M9LgPI,wB;QAEkD,sB8L/O+B,O9L+OzB,IAAK,K8L/O  
X,G9L+OmB,KAAM,K8L/OM,C9L+O/B,C;O;KAFID,C;0FAGA,yB;MAAA,+B;M8L9OJ,0B;M9L8OI,wB;QAE

mD,sB8L7OgC,O9L6O1B,IAAK,K8L7OX,G9L6OoB,KAAM,K8L7OM,C9L6OhC,C;O;KAFnD,C;0EAGA,yB;M  
AAA,+B;M8L5OJ,0B;M9L4OI,mB;QAEiC,sB8L3OqB,OAAP,C9L2OR,S8L3Oe,C9L2OrB,C;O;KAFjC,C;gFAIA,  
Y;MASmC,gB;K;kFACnC,yB;M8LpPJ,4B;M9LoPI,mB;QASqC,O8L1PiD,Q9L0P5C,S8L1PY,G9L0PE,G8L1P8B,  
C;O;K9LiPtF,C;8EAUA,Y;MASiC,OAAK,SAAL,GAAiB,G;K;gFACID,yB;MAAA,WASqD,wB;MATrD,mB;QA  
SmC,OAAK,oBAAL,SAAK,CAAL,U;O;KATnC,C;kFAWA,Y;MAEqC,W;K;oFACrC,yB;MAAA,iC;M8LtRJ,4B;  
M9LsRI,mB;QASuC,uB8L5R+C,Q9L4RnC,S8L5RG,G9L4RW,G8L5RqB,C9L4R/C,C;O;KATvC,C;gFAUA,yB;  
MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,GAAtB,C;O;KATnC,C;kFAUA,yB;MAAA,WAS6D,wB;  
MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,UAN,C;O;KATrC,C;kFAWA,Y;MAMqC,OAp  
DC,SAAL,GAAiB,G;K;oFAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,G;K;+BA6DID,Y;MAAyC,OAAQ,CA7DX,  
SAAL,GAAiB,GA6DD,Y;K;+BAPvRd,Y;MAAA,c;MAGsG,qD;MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCA  
HtG,G;K;wEAwVA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;0EAWA,yB;MAAA,0B;  
MAAA,+B;MAAA,4B;QAW2C,sBAAW,OAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,0B;MAAA,+  
B;MAAA,4B;QAWyC,sBAAW,OAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,  
4B;QAW0C,sBAAW,OAAL,SAAK,SAAX,C;O;KAX1C,C;IiCxXA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjCm  
VP,UiCnVO,aAAQ,KAAR,CjCmVP,C;K;uCiCjVtC,wB;MAOI,aAAQ,KAAR,IAAiB,KjC2Oc,K;K;kFiCvOL,Y;M  
AAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4  
CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,  
OjC8TY,UiC9TY,aAAM,mBAAN,EAAM,2BAAN,OjC8TZ,C;;QiC9T0C,MAAM,2BAAuB,YAAM,WAA7B,C;K;  
;0CAGtF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,E  
AAS,OjCsNO,KiCtNhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QdioDvB,U;QADhB,IAAI  
,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UcjoD6B,  
2BdioDR,OcjoDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,EdioD/B,OnB/6CF,KiCINiC,C;;UdioD9C,IAAI,OAAJ,C;  
YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;McloDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KA  
AqB,C;K;;IA9CvD,sC;MAAA,oD;MACgC,uBAAK,cAAU,IAAV,CAAL,C;MADhC,Y;K;+;oCAPJ,Y;MAAA,OA  
KqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFA  
wDA,yB;MAAA,yC;MAWsC,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCiMV,K;S;O;MiC5MvC,6B;QAWI,  
OAAO,oBAAW,+BAAU,IAAV,GAAGB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;IhCrE7C,oB;M  
AEpB,4B;MAFiG,gB;K;IAEjG,0B;MAAA,8B;MACI,iBAGmC,SAAK,CAAL,C;MAEnC,iBAGmC,SAAK,EAAL,  
C;MAEnC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANbtC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;oGAsBA,yB;M  
DqRA,6B;MCRQA,8C;MAhBA,wB;QAM0D,OaiBQ,YAAy,IAAK,KAAjB,EAA6B,CDuQ5D,cCxRsC,KDwR5B,  
KAAL,GAAiB,GAAtB,CCvQ4D,MAA7B,C;O;KAvBIE,C;oGAQA,yB;MC8QA,6B;MDtQA,8C;MARA,wB;QAM  
2D,OASO,YAAy,IAAK,KAAjB,EAA6B,CCwQ5D,cDjRuC,KCiR7B,KAAL,GAAiB,KAAtB,CDxQ4D,MAA7B,  
C;O;KAFIE,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAOkE,mBAAy,IAAK,KAAjB,EAAuB,KAAM,KAA7B,C;O;  
KAPIE,C;oGASA,yB;MA0RA,kBAS6D,sB;MAT7D,+B;MiB3RA,gD;MjBCA,wB;QAM0D,OiBAS,aAAkB,CjB6  
RhD,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB7RgD,MAAIB,EjBAGB,KiBAc,KAA9B,C;O;KjBnNE,C;0FAQA,y  
B;MDoPA,6B;MCpPA,wB;QAEsD,OAMD,cAAK,IAAK,KAAK,GAAW,CDqP5C,cC3P6B,KD2PnB,KAAL,GAA  
iB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KARrD,C;0FAGA,yB;MCKPA,6B;MDIPA,wB;QAEuD,OAGF,cAAK,IAA  
K,KAAK,GAAW,CCsP5C,cDzP8B,KCyPpB,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KALrD,C;0FAG  
A,yB;MAAA,6B;MAAA,wB;QAEqD,qBAAK,IAAK,KAAK,GAAK,KAAM,KAAX,IAAf,C;O;KAFrD,C;0FAGA,  
yB;MAyQA,kBAS6D,sB;MAT7D,+B;MAzQA,wB;QAEuD,OiBAA,eAAW,CjBGR7B,eAAW,oBAAL,SAAK,CA  
AL,iBAAN,CiBhR6B,MAAK,KjBAI,KiBAO,KAAX,CAAhB,C;O;KjBFvD,C;4FAIA,yB;MDuOA,6B;MCvOA,w  
B;QAEuD,OAMD,cAAK,IAAK,KAAK,GAAY,CDwO9C,cC9O+B,KD8OrB,KAAL,GAAiB,GAAtB,CCxO8C,M  
AAZ,IAAf,C;O;KARtD,C;4FAGA,yB;MCqOA,6B;MDrOA,wB;QAEwD,OAGF,cAAK,IAAK,KAAK,GAAY,CCy  
O9C,cD5OgC,KC4OtB,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MA  
AA,wB;QAEsD,qBAAK,IAAK,KAAK,GAAM,KAAM,KAAX,IAAf,C;O;KAFtD,C;4FAGA,yB;MA4PA,kBAS6D  
,sB;MAT7D,+B;MA5PA,wB;QAEwD,OiBAA,eAAW,CjBmQ9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBnQ8B  
,MAAK,UjBAK,KiBAO,KAAX,CAAhB,C;O;KjBFxD,C;4FAIA,yB;MD0NA,6B;MC1NA,wB;QAEuD,OAMD,cA  
Ae,YAAV,IAAK,KAAK,EAAY,CD2N9C,cCjO+B,KDiOrB,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;

KARtD,C;4FAGA,yB;MCwNA,6B;MDxNA,wB;QAEwD,OAGF,cAAe,YAAV,IAAK,KAAC,EAAY,CC4N9C,cD /NgC,KC+NtB,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB; QAEsD,qBA Ae,YAAV,IAAK,KAAC,EAAM,KAAM,KA AZ,CAAf,C;O;KAFtD,C;4FAGA,yB;MA+OA,kBAS6D ,sB;MAT7D,+B;MA/OA,wB;QAEwD,OiBAA,eAAW,CjBsP9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBtP8B, MAAK,UjBAK,KiBAO,KAAZ,CAAhB,C;O;KjBFxD,C;wFAIA,yB;MD6MA,6B;MCvMA,4C;MANA,wB;QAEq D,OAMD,WAAW,IAAX,ED8MjB,cCpN2B,KDoNjB,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KARpD,C;wFAGA,y B;MC2MA,6B;MDxMA,4C;MAHA,wB;QAEsD,OAGF,WAAW,IAAX,EC+MjB,cDIN4B,KCKNIB,KAAL,GAAi B,KAAtB,CD/MiB,C;O;KALpD,C;wFAGA,yB;MAAA,4C;MAAA,wB;QAEoD,kBAAW,IAAX,EAAiB,KAAjB,C ;O;KAFpD,C;wFAGA,yB;MAkOA,kBAS6D,sB;MAT7D,+B;MiBIOA,8C;MjBAA,wB;QAEsD,OiBAA,YjByOjB, eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBzOiB,EjBAmB,KiBanB,C;O;KjBFtD,C;wFAIA,yB;MDgMA,6B;MCIL A,kD;MAdA,wB;QAMqD,OAcD,cAAc,IAAd,EDqLjB,cCnM2B,KDmMjB,KAAL,GAAiB,GAAtB,CCrLiB,C;O;K ApBpD,C;wFAOA,yB;MC0LA,6B;MDnLA,kD;MAPA,wB;QAMsD,OAOF,cAAc,IAAd,ECsLjB,cD7L4B,KC6Ll B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KAbpD,C;wFAOA,yB;MAAA,kD;MAAA,wB;QAMoD,qBAAc,IAAd,EA AoB,KAApB,C;O;KANpD,C;wFAOA,yB;MAyMA,kBAS6D,sB;MAT7D,+B;MiBzMA,oD;MjBAA,wB;QAMsD, OiBAA,ejB4MjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB5MiB,EjBAmB,KiBanB,C;O;KjBNtD,C;kGAQA,yB; MDmKA,6B;MCvMA,4C;MAoCA,wB;QAMiD,OAxCg,WAAW,IAAX,ED8MjB,cCtK4B,KDsKIB,KAAL,GAAi B,GAAtB,CC9MiB,C;O;KakCpD,C;kGAOA,yB;MC6JA,6B;MDxMA,4C;MA2CA,wB;QAMkD,OA/CE,WAAW, IAAX,EC+MjB,cDhK6B,KCgKnB,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KAYCpD,C;kGAOA,yB;MAIDA,4C;MA kDA,wB;QAMgD,OAtDI,WAAW,IAAX,EAsDA,KatDA,C;O;KAgDpD,C;kGAOA,yB;MA4KA,kBAS6D,sB;MA T7D,+B;MiBIOA,8C;MjBsDA,wB;QAMkD,OiB1DI,YjByOjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBzOiB,Ej B0DoB,KiB1DpB,C;O;KjBoDtD,C;wFAQA,yB;MDsIA,6B;MCILA,kD;MDiPJ,0B;MAAA,+B;MCRMI,wB;QAQ6 C,ODwMR,eAAW,OCtPI,cAAc,IAAd,EDqLjB,cCvImB,KDuIT,KAAL,GAAiB,GAAtB,CCrLiB,CA4Lf,KD0DW, CAAX,C;O;KChNrC,C;wFASA,yB;MC8HA,6B;MDnLA,kD;MCKPJ,4B;MAAA,iC;MD7LI,wB;QAQ+C,OCgMR, gBAAY,QDvPC,cAAc,IAAd,ECsLjB,cD/HqB,KC+HX,KAAL,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C; O;KDxMvC,C;wFASA,yB;MA9DA,kD;MA8DA,wB;QAQ2C,OAhES,cAAc,IAAd,EA gEL,KAhEK,C;O;KAwDp D,C;wFASA,yB;MAyIA,kBAS6D,sB;MAT7D,+B;MiBzMA,oD;MjBgEA,wB;QAQ6C,OiBIES,ejB4MjB,eAAW,o BAAL,SAAK,CAAL,iBAAN,CiB5MiB,EjBkEU,KiBIEV,C;O;KjB0DtD,C;wEAUA,yB;MAAA,6B;MAAA,mB;Q AMyC,qBAAK,SAAK,QAAV,C;O;KANzC,C;wEAQA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QAA V,C;O;KANzC,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAE6D,0BAAU,IAAV,EAAGB,KAAhB,C;O;KAF7D,C;S GAIA,yB;MAAA,yC;MAAA,wB;QAQgE,mBAAW,KAAX,C;O;KARhE,C;wFAUA,yB;MAAA,6B;MAAA,2B;QA OmD,qBAAK,aAAS,QAAc,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,2B;QAOMD,qBAAK,cAAU,QAAf,C; O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAC,KAAM,KAazB,C;O;KAFj D,C;SFAGA,yB;MAAA,6B;MAAA,wB;QAEgD,qBAAK,IAAK,KAAL,GAAa,KAAM,KAaxB,C;O;KAFhD,C;wF AGA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAC,KAAM,KAazB,C;O;KAFjD,C;wEAGA,y B;MAAA,6B;MAAA,mB;QAEgC,qBAAU,CAAL,SAAL,C;O;KAFhC,C;8EAIA,yB;MAAA,0B;MAAA,mB;QAU mC,OAAK,OAAL,SAAK,C;O;KAVxC,C;gFAWA,yB;MAAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK,C;O ;KAVIC,C;4EAWA,Y;MASiC,gB;K;8EACjC,yB;MAAA,kBASqD,sB;MATrD,mB;QASmC,OAAK,oBAAL,SA A K,CAAL,iB;O;KATnC,C;gFAWA,yB;MDwDJ,0B;MAAA,+B;MCxDI,mB;QASqC,OD0DA,eAAW,OC1DX,SD0 DW,CAAX,C;O;KCnErC,C;kFAUA,yB;MC+CJ,4B;MAAA,iC;MD/CI,mB;QASuC,OCiDA,gBAAY,QDjDZ,SCiD Y,CAAZ,C;O;KD1DvC,C;8EAUA,Y;MAEmC,W;K;gFACnC,yB;MAAA,kBAS6D,sB;MAT7D,+B;MAAA,mB;Q ASqC,sBAAW,oBAAL,SAAK,CAAL,iBAAN,C;O;KATrC,C;gFAWA,yB;MASA,gD;MATA,mB;QAQqC,OA OE,aAAa,SAAb,C;O;KafvC,C;kFASA,yB;MAAA,gD;MAAA,mB;QAMuC,oBAAa,SAAb,C;O;KANvC,C;8BAQA,Y ;MAAyC,OArDD,oBAAL,SAAK,CAAL,iBAqDe,W;K;8BA1WiD,Y;MAAA,c;MAGqG,qD;MAHrG,a;K;4BAA A,iB;MAAA,2IAGqG,oCAHrG,G;K;sEA8WA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAU,SAAV,C;O;KAXxC, C;wEAYA,yB;MAAA,6B;MAAA,4B;QAWyC,qBAAU,SAAV,C;O;KAXzC,C;wEAYA,yB;MAAA,6B;MAAA,4 B;QAUuC,qBAAK,SAAL,C;O;KAVvC,C;wEAWA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAK,SAAK,QAAV,C ;O;KAXxC,C;uEAaA,yB;MAAA,gD;MAAA,4B;QASyC,oBAAkB,SAaIB,C;O;KATzC,C;wEAUA,yB;MAAA,gD ;MAAA,4B;QAS0C,oBAAa,SAAb,C;O;KAT1C,C;IiCraA,4B;MACqB,sB;K;sCAKjB,iB;MAM4C,OjCiYT,SiCjYS

,aAAQ,KAAR,CjCiYT,C;K;sCiC/XnC,wB;MAOI,aAAQ,KAAR,IAAiB,KjCmRY,K;K;iFiC/QH,Y;MAAQ,OAAA  
,YAAQ,O;K;mCAE9C,Y;MAC6E,8BAAS,YAAT,C;K;IAEvD,mC;MAAC,oB;MACnB,eAAoB,C;K;2CACpB,Y;  
MAAyB,sBAAQ,YAAM,O;K;wCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjC4WS,S  
iC5We,aAAM,mBAAN,EAAM,2BAAN,OjC4Wf,C;;QiC5W4C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;yCAGRf,  
mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,K;MAEvC,OAAe,WAAR,YAAQ,EAAS,OjC  
8PK,KiC9Pd,C;K;8CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QfioDvB,U;QADhB,IAAI,wCAAsB,  
mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UejoD6B,2BfioDR,  
OejoDQ,O;UAAA,W;YAAsB,oBAAR,YAAQ,EfioD9B,OIBv4CJ,KiC1PkC,C;;UfioD7C,IAAI,OAAJ,C;YAAyB,a  
AAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MeloDH,iB;K;kCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;  
IA9CvD,qC;MAAA,mD;MACgC,sBAAK,eAAS,IAAT,CAAL,C;MADhC,Y;K;;;mCAPJ,Y;MAAA,OAKqB,oDA  
LrB,M;K;mCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;iCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;8EAwDA,yB;  
MAAA,uC;MAWoC,wC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCyOV,K;S;O;MiCpPrC,6B;QAWI,OAAO,m  
BAAU,gCAAS,IAAT,GAAe,sBAAf,CAAV,C;O;KAXX,C;gFAcA,oB;MAGkE,e;K;IkMjE5C,wC;MA8BIB,iC;MA  
9BsD,2BAAgB,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;  
yFAKR,Y;MACxB,Q;MAAJ,IAAI,yCAAQ,4BAAK,UAAb,QAAJ,C;QvNmHyC,MAAM,6BuNnHb,6EvNmH2C,  
WAA9B,C;;MuNIH/C,OnOoDiD,SmOpD1C,SnOoDoD,KAAK,GAAW,CmOpD7D,WnOoD6D,MAAX,IAAf,C;K;  
2CmOjDrD,iB;MAA8C,WnO+BoB,YmO/BpB,UnO+BqC,KAAjB,EmO/BX,KnO+BwC,KAA7B,CmO/BpB,K;M  
AAA,S;QAAkB,OnO+BE,YmO/BF,KnO+BmB,KAAjB,EmO/BO,SnO+Bsb,KAA7B,CmO/BF,K;;MAAIB,W;K;k  
CAE9C,Y;MAKkC,OnOwBgC,YmOxBhC,UnOwBiD,KAAjB,EmOxBxB,SnOwBqD,KAA7B,CmOxBhC,I;K;iCA  
EIC,iB;MAEY,UAAwB,M;MADhC,2CAAuB,kBAAa,KAAM,UAAAnB,KACf,2CAAS,KAAM,MAAf,cAAwB,6C  
AAQ,KAAM,KAAAd,QAAxB,CADe,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,U  
nO0QA,KmO1QL,QAAqB,SnO0QhB,KmO1QL,I;K;mCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,+B;  
MAAA,mC;MACI,aAC8B,cAAU,4BAAK,UAAf,EAA0B,4BAAK,UAA/B,C;K;;IAFIC,2C;MAAA,0C;QAAA,yB  
;;MAAA,mC;K;;IAyJ,oD;MA4CI,uC;MATCI,IAAI,SAAQ,CAAZ,C;QAAuB,MAAA,gCAAyB,wBAAzB,C;MACp  
C,IAAI,SAAQ,WAAZ,C;QAA2B,MAAA,gCAAyB,wEAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGwB,4BAA0B,K  
AA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YAGuB,I;K;yCAEvB,Y;MAAgD,mCAAwB,UAAxB,EAA+B,S  
AA/B,EAAqC,SAArC,C;K;wCAEhD,Y;MAMqC,OAAI,YAAO,CAAX,GnOhC6B,YmOgCf,UnOhCgC,KAAjB,E  
mOgCP,SnOhCoC,KAA7B,CmOgCf,IAAd,GnOhC6B,YmOgCG,UnOhCc,KAAjB,EmOgCW,SnOhCkB,KAA7B,  
CmOgCG,I;K;uCAErE,iB;MAEY,UAAwB,M;MADhC,iDAA6B,kBAAa,KAAM,UAAAnB,KACrB,2CAAS,KAAM  
,MAAf,cAAwB,6CAAQ,KAAM,KAAAd,QAAxB,KAA8C,cAAQ,KAAM,KADvC,CAA7B,C;K;yCAGJ,Y;MACI,O  
AAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UnOkNN,KmOINC,QAAqB,SnOkNtB,KmOINC,IAAN,SAAGD  
,SAAhD,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GAAGD,  
UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAg  
B,UAAhB,EAA4B,QAA5B,EAA5C,IAAtC,C;K;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAmBiC,oD;M  
ACjC,sBAA2B,I;MAC3B,iBAAmC,OAAO,CAA1C,GnOhEkE,YmOgErB,KnOhEsC,KAAjB,EmOgEZ,InOhEyC,  
KAA7B,CmOgErB,KAA7C,GnOhEkE,YmOgEF,KnOhEmB,KAAjB,EmOgEO,InOhEsB,KAA7B,CmOgEF,K;M  
AChE,cnO6RmC,SmO7RhB,InO6RgB,C;MmO5RnC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAE3C,Y;MA  
AkC,qB;K;6CAEIC,Y;MACI,YAAy,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UA  
Ac,MAAA,6B;QAC3B,iBAAU,K;;QAEV,cnO1D6C,SmO0D7C,WnO1DuD,KAAK,GmO0DpD,WnO1D+D,KAA  
X,IAAf,C;;MmO4DjD,OAAO,K;K;;IIN7HU,qB;MAErB,6B;MAFkG,gB;K;IAEIG,2B;MAAA,+B;MACI,iBAGoC,  
a;MAEpC,iBAGoC,c;MAEpC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANbtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B  
;K;sGAsBA,yB;MIB+RA,WAS6D,wB;MAT7D,+B;MkBVQA,gD;MAxBA,wB;QAM0D,OAYBS,aAAa,IAAK,KA  
AIB,EAA8B,CIByQ5D,eAAW,oBkBIyB,KIBkS9B,KAAK,CAAL,UAAAN,CkBzQ4D,MAA9B,C;O;KA/BnE,C;sG  
AQA,yB;MhBwRA,aAS6D,0B;MAT7D,+B;MgBxQA,gD;MAhBA,wB;QAM2D,OAIbQ,aAAa,IAAK,KAAIB,EA  
A8B,ChB0Q5D,eAAW,oBgB3R0B,KhB2R/B,KAAK,CAAL,YAAN,CgB1Q4D,MAA9B,C;O;KAvBnE,C;sGAQA  
,yB;MjBmSA,kBAS6D,sB;MAT7D,+B;MiB3RA,gD;MARA,wB;QAMyD,OASU,aAAa,IAAK,KAAIB,EAA8B,Cj  
B6R5D,eAAW,oBiBtSwB,KjBsS7B,KAAK,CAAL,iBAAN,CiB7R4D,MAA9B,C;O;KafnE,C;kGAQA,yB;MAAA  
,gD;MAAA,wB;QAomE,oBAAa,IAAK,KAAIB,EAAwB,KAAM,KAA9B,C;O;KAPnE,C;4FASA,yB;MIB8PA,W



AS6D,wB;MAT7D,+B;MkB9PA,wB;QAEuD,OASA,eAAM,IAAK,KAAK,KAAW,CIB4P7C,eAAW,oBkBrQiB,K  
lBqQtB,KAAK,CAAL,UAAN,CkB5P6C,MAAX,CAAhB,C;O;KAXvD,C;4FAGA,yB;MhB4PA,aAS6D,0B;MAT7  
D,+B;MgB5PA,wB;QAEwD,OAMD,eAAM,IAAK,KAAK,KAAW,ChB6P7C,eAAW,oBgBnQkB,KhBmQvB,KA  
AK,CAAL,YAAN,CgB7P6C,MAAX,CAAhB,C;O;KARvD,C;4FAGA,yB;MjB4QA,kBAS6D,sB;MAT7D,+B;MiB  
5QA,wB;QAEsD,OAGC,eAAM,IAAK,KAAK,KAAW,CjBgr7C,eAAW,oBiBnRgB,KjBmRrB,KAAK,CAAL,iBA  
AN,CiBhR6C,MAAX,CAAhB,C;O;KALvD,C;4FAGA,yB;MAAA,+B;MAAA,wB;QAEuD,sBAAM,IAAK,KAAK  
,KAAK,KAAW,KAAX,CAAhB,C;O;KAFvD,C;8FAIA,yB;MIBiPA,WAS6D,wB;MAT7D,+B;MkBjPA,wB;QAEw  
D,OASA,eAAM,IAAK,KAAK,UAAY,CIB+O/C,eAAW,oBkBxPmB,KIBwPxB,KAAK,CAAL,UAAN,CkB/O+C,  
MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;MhB+OA,aAS6D,0B;MAT7D,+B;MgB/OA,wB;QAEyD,OAMD,eAA  
M,IAAK,KAAK,UAAY,ChBgP/C,eAAW,oBgBtPoB,KhBsPzB,KAAK,CAAL,YAAN,CgBhP+C,MAAZ,CAAhB,  
C;O;KARxD,C;8FAGA,yB;MjB+PA,kBAS6D,sB;MAT7D,+B;MiB/PA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,  
UAAY,CjBmQ/C,eAAW,oBiBtQkB,KjBsQvB,KAAK,CAAL,iBAAN,CiBnQ+C,MAAZ,CAAhB,C;O;KALxD,C;8  
FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,sBAAM,IAAK,KAAK,UAAM,KAAZ,CAAhB,C;O;KAFxD,C  
;8FAIA,yB;MIBoOA,WAS6D,wB;MAT7D,+B;MkBpOA,wB;QAEwD,OASA,eAAM,IAAK,KAAK,UAAY,CIBk  
O/C,eAAW,oBkB3OmB,KIB2OxB,KAAK,CAAL,UAAN,CkBio+C,MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;  
MhBkOA,aAS6D,0B;MAT7D,+B;MgBIOA,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAY,ChBmO/C,eAAW,o  
BgBzOoB,KhByOzB,KAAK,CAAL,YAAN,CgBnO+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MjBkPA,kBA  
S6D,sB;MAT7D,+B;MiBIPA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,UAAY,CjBsP/C,eAAW,oBiBzPkB,KjByP  
vB,KAAK,CAAL,iBAAN,CiBtP+C,MAAZ,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,s  
BAAM,IAAK,KAAK,UAAM,KAAZ,CAAhB,C;O;KAFxD,C;0FAIA,yB;MIBuNA,WAS6D,wB;MAT7D,  
+B;MkB9MA,8C;MATA,wB;QAEsD,OASA,YAAY,IAAZ,EIBqNjB,eAAW,oBkB9Ne,KIB8NpB,KAAK,CAAL,  
UAAN,CkBrNiB,C;O;KAXtD,C;0FAGA,yB;MhBqNA,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MANA,wB;QAEuD,  
OAMD,YAAY,IAAZ,EhBsNjB,eAAW,oBgB5NgB,KhB4NrB,KAAK,CAAL,YAAN,CgBtNiB,C;O;KARtD,C;0F  
AGA,yB;MjBqOA,kBAS6D,sB;MAT7D,+B;MiBIOA,8C;MAHA,wB;QAEqD,OAGC,YAAY,IAAZ,EjByOjB,eA  
AW,oBiB5Oc,KjB4OnB,KAAK,CAAL,iBAAN,CiBzOiB,C;O;KALtD,C;0FAGA,yB;MAAA,8C;MAAA,wB;QAE  
sD,mBAA,YIAAZ,EAakB,KAAIB,C;O;KAFtD,C;0FAIA,yB;MIB0MA,WAS6D,wB;MAT7D,+B;MkBrlA,oD;M  
ArBA,wB;QAMsD,OAqBA,eAAe,IAAf,EIBwLjB,eAAW,oBkB7Me,KIB6MpB,KAAK,CAAL,UAAN,CkBxLiB,C  
;O;KA3BtD,C;0FAOA,yB;MhBoMA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MAdA,wB;QAMuD,OAcd,eAAe,IAAf  
,EhByLjB,eAAW,oBgBvMgB,KhBuMrB,KAAK,CAAL,YAAN,CgBzLiB,C;O;KApBtD,C;0FAOA,yB;MjBgNA,k  
BAS6D,sB;MAT7D,+B;MiBzMA,oD;MAPA,wB;QAMqD,OAOC,eAAe,IAAf,EjB4MjB,eAAW,oBiBnNc,KjBmN  
nB,KAAK,CAAL,iBAAN,CiB5MiB,C;O;KAbtD,C;0FAOA,yB;MAAA,oD;MAAA,wB;QAMsD,sBAae,IAAf,EA  
AqB,KAArB,C;O;KANtD,C;oGAQA,yB;MIB6KA,WAS6D,wB;MAT7D,+B;MkB9MA,8C;MAiCA,wB;QAMkD,  
OArCI,YAAY,IAAZ,EIBqNjB,eAAW,oBkBhLgB,KIBgLrB,KAAK,CAAL,UAAN,CkBrNiB,C;O;KA+BtD,C;oG  
AOA,yB;MhBuKA,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MAwCA,wB;QAMmD,OA5CG,YAAY,IAAZ,EhBsNjB,  
eAAW,oBgB1KiB,KhB0KiB,KAAK,CAAL,YAAN,CgBtNiB,C;O;KAsCtD,C;oGAOA,yB;MjBmLA,kBAS6D,sB;  
MAT7D,+B;MiBIOA,8C;MA+CA,wB;QAMiD,OAnDK,YAAY,IAAZ,EjByOjB,eAAW,oBiBtLe,KjBsLpB,KAAK  
,CAAL,iBAAN,CiBzOiB,C;O;KA6CtD,C;oGAOA,yB;MatDA,8C;MASDA,wB;QAMkD,OA1DI,YAAY,IAAZ,E  
A0DA,KA1DA,C;O;KAoDtD,C;0FAQA,yB;MIBgJA,WAS6D,wB;MAT7D,+B;MkBrlA,oD;MIBsPJ,0B;MAAA,+  
B;MkBjNI,wB;QAQ6C,OiBoNP,eAAW,OkB3PK,eAAe,IAAf,EIBwLjB,eAAW,oBkBjJM,KIBiJX,KAAK,CAAL,  
UAAN,CkBxLiB,CAsLjB,KIBqEY,SAAX,C;O;KkB5NtC,C;0FASA,yB;MhBwIA,aAS6D,0B;MAT7D,+B;MgBtL  
A,oD;MhBuPJ,4B;MAAA,iC;MgBzMI,wB;QAQ+C,OhB4MP,gBAAY,QgB5PE,eAAe,IAAf,EhByLjB,eAAW,oBg  
BzIQ,KhByIb,KAAK,CAAL,YAAN,CgBzLiB,CAGMf,KhB4Da,SAAZ,C;O;KgBpNx,C;0FASA,yB;MjBkJA,kB  
AS6D,sB;MAT7D,+B;MiBzMA,oD;MjB4QJ,6B;MiBrNI,wB;QAQ2C,OjBwNP,ciBjRkB,eAAe,IAAf,EjB4MjB,eA  
AW,oBiBnJI,KjBmJT,KAAK,CAAL,iBAAN,CiB5MiB,CA0MnB,KjBuEW,QAAV,C;O;KiBhOpC,C;0FASA,yB;  
MAhEA,oD;MAGEA,wB;QAQ6C,OAIES,eAAe,IAAf,EAkEL,KAIEK,C;O;KA0DtD,C;0EAUA,yB;MAAA,+B;M  
AAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,  
SAAK,MAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,gD;MAAA,wB;QAE+D,2BAAW,IAAX,EAaiB,KAAjB,C;O;  
KAF/D,C;wGAIA,yB;MAAA,yC;MAAA,wB;QAQkE,mBAAW,KAAX,C;O;KARIE,C;0FAUA,yB;MAAA,+B;M

AAA,2B;QAOoD,sBAAM,oBAAS,QAAT,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,2B;QAOoD,sBAAM,6BAAU,QAAV,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAc,KAAM,KAAPB,CAAN,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;MAAA,wB;QAEkD,sBAAM,IAAK,KAAL,IAAa,KAAM,KAAnB,CAAN,C;O;KAFID,C;0FAGA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAC,KAAM,KAAPB,CAAN,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;MAAA,mB;QAEiC,sBAAM,SAAK,MAAX,C;O;KAFjC,C;gFAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,S;O;KAVx,C;KFAWA,yB;MAAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK,S;O;KAV1C,C;8EAWA,Y;MAUic,OAAA,SAAK,Q;K;gFAcTc,Y;MASmC,gB;K;kFAEnC,yB;MIBmEJ,0B;MAAA,+B;MkBnEI,mB;QASqC,OIBqEC,eAAW,OkBrEZ,SIBqEY,SAAX,C;O;KkB9EtC,C;oFAUA,yB;MhB0DJ,4B;MAAA,iC;MgB1DI,mB;QASuC,OhB4DC,gBAAY,QgB5Db,ShB4Da,SAAZ,C;O;KgBrExC,C;gFAUA,yB;MjBqEJ,6B;MiBrEI,mB;QASmC,OjBuEC,ciBvED,SjBuEW,QAAV,C;O;KiBhFpC,C;kFAUA,Y;MAEqC,W;K;kFAErC,yB;MASA,kD;MATA,mB;QAQqC,OASE,cAAc,SAAd,C;O;KAjBvC,C;oFASA,yB;MAAA,kD;MAAA,mB;QAQuC,qBAAc,SAAd,C;O;KARvC,C;+BAUA,Y;MAAyC,qBAAc,SAAd,C;K;+;+BA7W7C,Y;MAAA,c;MAGsG,qD;MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCAHtG,G;K;wEaiXA,yB;MAAA,+B;MAAA,4B;QAW0C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX1C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAW2C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAWyC,sBAAW,oBAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;yEAYA,yB;MAAA,kD;MAAA,4B;QAS2C,qBAAmB,SAAnB,C;O;KAT3C,C;0EAUA,yB;MAAA,kD;MAAA,4B;QAS4C,qBAAc,SAAd,C;O;KAT5C,C;IiBxaA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjBgZP,UiBhZO,aAAQ,KAAR,CjBgZP,C;K;uCiB9YtC,wB;MAOI,aAAQ,KAAR,IAAiB,KjB8Rc,K;K;kFiB1RL,Y;MAAQ,OAA A,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjB2XY,UiB3XY,aAAM,mBAAN,EAAM,2BAAN,OjB2XZ,C;;QiB3X0C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;0CAGtF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OjByQO,KiBzQhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QhBioDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAaHb,OAAgB,gBAaHb,C;UAAgB,2B;UgBjoD6B,2BhBioDR,OgBjoDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,EhBioD/B,OD53CF,KiBrQiC,C;;UhBioD9C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MgBloDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA9CvD,sC;MAAA,oD;MACgC,uBAAK,iBAAU,IAAV,CAAL,C;MADhC,Y;K;;oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAwDA,yB;MAAA,yC;MAWSc,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjBoPV,K;S;O;MiB/PvC,6B;QAWI,OAAO,oBAAW,kBAAU,IAAV,EAAGB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;IkMjE9C,2C;MA8BnB,kC;MA9ByD,4BAAiB,KAAjB,EAawB,YAAxB,K;K;qFAC/B,Y;MAAQ,iB;K;4FACD,Y;MAAQ,gB;K;4FAKR,Y;MACzB,Q;MAAJ,IAAI,yCAAQ,6BAAM,UAAAd,QAAJ,C;QxNmHyC,MAAM,6BwNnHZ,6ExNmH0C,WAA9B,C;;MwNIH/C,OnNuDmD,UmNvD5C,SnNuDuD,KAAC,KAAW,CjBgR7C,UAAW,oBAAL,CoOvUzB,WpOuUyB,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,C;K;8CmNpDvD,iB;MAA+C,WnNuCoB,amNvCpB,UnNuCsC,KAAIB,EmNvCX,KnNuCyC,KAA9B,CmNvCpB,K;MAAA,S;QAAkB,OnNuCE,amNvCF,KnNuCoB,KAAIB,EmNvCO,SnNuCuB,KAA9B,CmNvCF,K;;MAAIB,W;K;qCAE/C,Y;MAKkC,OnNgCiC,amNhCjC,UnNgCmD,KAAIB,EmNhCzB,SnNgCuD,KAA9B,CmNhCjC,I;K;oCAEIC,iB;MAEY,UAAwB,M;MADhC,8CAAwB,kBAAa,KAAM,UAAAnB,KACHB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,CADgB,CAAxB,C;K;sCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,MnN2QK,CArCkB,UmNtOjB,UnNsO4B,KAAL,KAAoB,CAVzB,UmN5NP,UnN4Na,yBmN5NH,EnN4NG,CAAN,CAUyB,MAAPB,CAAN,CAqCIB,MAAK,QmN3QV,QnN2QK,CArCkB,UmNtOoB,SnNsOT,KAAL,KAAoB,CAVzB,UmN5N6B,SnN4NvB,yBmN5NgC,EnN4NhC,CAN,CAUyB,MAAPB,CAAN,CAqCIB,MAAK,QmN3QV,I;K;sCAE5B,Y;MAAkC,OAAE,UAAf,qBAAU,S;K;IAE5C,gC;MAAA,oC;MACI,aAC+B,iBAAW,6BAAM,UAAjB,EAA4B,6BAAM,UAAIC,C;K;;IAFnC,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;;IAYJ,qD;MA4CI,wC;MATCI,IAAI,gBAAJ,C;QAAwB,MAAa,gCAAYB,wBAAzB,C;MACrC,IAAI,sCAAJ,C;QAA4B,MAAa,gCAAYB,yEAAzB,C;MAG7C,aAG0B,K;MAE1B,YAGyB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEZB,YAGwB,I;K;0CAExB,Y;MAAiD,oCAAYB,UAAzB,EAAgC,SAAhC,EAAcS,SAATC,C;K;yCAEjD,Y;MAMqC,OAAI,uBAAO,CAAX,GnNx8B8B,amNwBhB,UnNx8kC,KAAIB,Em

NwBR,SnNxBsC,KAA9B,CmNwBhB,IAAd,GnNx8B,amNwBE,UnNxBgB,KAA1B,EmNwBU,SnNxBoB,KAA9B,CmNwBE,I;K;wCAErE,iB;MAEY,UAAwB,M;MADhC,kDAA8B,kBAAa,KAAM,UAAAnB,KACtB,2CAAS,KAAm,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAXB,KAA8C,kBAAQ,KAAM,KAAd,CADxB,CAA9B,C;K;0CAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,OAAM,MnNmND,CArCkB,UmN9KX,UnN8KsB,KAAAL,KAAoB,CAVzB,UmNpKD,UnNoKO,yBmNpKG,EnNoKH,CAAN,CAUyB,MAApB,CAAN,CAqCIB,MAAK,QmNnNJ,QnNmND,CArCkB,UmN9K0B,SnN8Kf,KAAAL,KAAoB,CAVzB,UmNpKMc,SnNoK7B,yBmNpKsC,EnNoKtC,CAAN,CAUyB,MAApB,CAAN,CAqCIB,MAAK,QmNnNJ,IAAN,SAAqF,cAAU,6BAAU,EAAV,CAA V,CAAyB,QAA9G,I;K;0CAE5B,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAf,qBAAU,SAAV,cAAqB,SAArB,WAAAd,GAAgD,UAAf,2BAAgB,SAAhB,cAA6B,SAAD,aAA5B,W;K;IAEHf,sC;MAAA,0C;K;mEACI,sC;MAQ+F,4BAAiB,UAAjB,EAA6B,QAA7B,EAAuC,IAAvC,C;K;;;IATnG,kD;MAAA,iD;QAAA,gC;;MAAA,0C;K;;IAmBkC,qD;MACIC,sBAA2B,I;MAC3B,iBAAmC,kBAAO,CAA1C,GnNxDmE,amNwDtB,KnNxDwC,KAA1B,EmNwDb,InNx2D2C,KAA9B,CmNwDtB,KAA7C,GnNxDmE,amNwDH,KnNxDqB,KAA1B,EmNwDM,InNxDwB,KAA9B,CmNwDH,K;MACHe,cnN4SsC,UmN5SnB,InN4SmB,C;MmN3StC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;iDAE3C,Y;MAAkC,qB;K;8CAEIC,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAA,6B;QAC3B,iBAAU,K;;QAEV,cnNvD+C,UmNuD/C,WnNvD0D,KAAK,KmNuDvD,WnNvDkE,KAAAX,CAAhB,C;;MmNyDnD,OAAO,K;K;;wECrIf,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;oFC7BA,yB;MAAA,gD;MAAA,4B;QAM6C,OAAQ,atOyShB,csOzSgB,C;O;KANrD,C;oGAQA,yB;MtHwCA,4B;MsHxCA,4B;QAMqD,OtHwCM,YhHyPtB,cgHzPsB,C;O;KsH9C3D,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAQ,sBtOyRzB,csOzRyB,C;O;KAN9D,C;8FAQA,yB;MAAA,0D;MtOkXA,6B;MsOIxA,4B;QAOmD,OtOqXZ,csOrXoB,kBtOgRtB,csOhRsB,CtOqXpB,C;O;KsO5XvC,C;4FASA,yB;MAAA,wD;MtOyWA,6B;MsOzWA,4B;QAOkD,OtO4WX,csO5WmB,iBtOuQrB,csOvQqB,CtO4WnB,C;O;KsOnXvC,C;gFASA,yB;MAAA,4C;MtOgWA,6B;MsOhWA,sC;QAayD,OtO6V1B,csO7V0B,WtOwP5B,csOxP4B,EAAW,QAAX,CtO6V1B,C;O;KsO1WvC,C;kFAgBA,yB;MAAA,8C;MtOgVA,6B;MsOhVA,sC;QAa0D,OtO6UnB,csO7U2B,YtOwO7B,csOxO6B,EAA Y,QAAX,CtO6U3B,C;O;KsO1VvC,C;oFAgBA,yB;MAAA,gD;MAAA,4B;QAM8C,OAAS,arN0OhB,cqN1OgB,C;O;KANvD,C;oGAQA,yB;MAAA,gE;MAAA,4B;QAMsD,OAAS,qBrNkOxB,cqN1OwB,C;O;KAN/D,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMuD,OAAS,sBrN0NzB,cqN1NyB,C;O;KANhE,C;8FAQA,yB;MAAA,0D;MrNuTA,+B;MqNvTA,4B;QAOqD,OrN0TX,eqN1Tob,kBrNiNvB,cqNjNuB,CrN0TpB,C;O;KqNjU1C,C;4FASA,yB;MAAA,wD;MrN8SA,+B;MqN9SA,4B;QAOoD,OrNiTV,eqNjTmB,iBrNwMtB,cqNxMsB,CrNiTnB,C;O;KqNxT1C,C;+EASA,yB;MAAA,4C;MrNqSA,+B;MqNrSA,sC;QAa2D,OrNkSjB,eqN1S0B,WrNyL7B,cqNzL6B,EAAW,QAAX,CrNkS1B,C;O;KqN/S1C,C;iFAeA,yB;MtHgEA,4C;M/FsNA,+B;MqNtRA,sC;QAa4D,OrNmRIB,e+FnNuB,W/F0G1B,c+F1G0B,EAAW,CsHhEK,QtHgEL,IAAX,C/FmNvB,C;O;KqNhS1C,C;oFAeA,yB;MvOkKI,6B;MuOpTJ,gD;MAKJA,4B;QAM8C,OAIJO,atOyShB,CDcE,cAAU,cAAL,GAAiB,GAAtB,CCdF,MsOzSgB,C;O;KA4IrD,C;oGAQA,yB;MtH1GA,4B;MsH0GA,4B;QAMsD,OtH1GK,YjHiNpB,ciOpHe,GAAY,GhH7FP,CgH6FN,GAA6C,EAA7C,I;O;KMOrD,C;sGAQA,yB;MNbA,kE;MMaA,4B;QAMuD,ONbkB,sBjO4GIC,ciO5GgB,GAAW,GAAO,C;O;KMOzE,C;8FAQA,yB;MAAA,0D;MvOyMA,0B;MAAA,+B;MuOzMA,4B;QAOqD,OvO6MZ,eAAW,OuO7MS,kBvO0GnB,cAAL,GAAiB,GuO1GO,CvO6MT,C AAX,C;O;KuOpNzC,C;4FASA,yB;MAAA,wD;MvOgMA,0B;MAAA,+B;MuOhMA,4B;QAOoD,OvOoMX,eAAW,OuOpMQ,iBvOiGIB,cAAL,GAAiB,GuOjGM,CvOoMR,CAAX,C;O;KuO3MzC,C;gFAUA,yB;MAAA,4C;MvO+JA,+B;MuO/JA,sC;QAa2D,OvO4JjB,euO5J0B,WvO6D7B,cuO7D6B,EAAW,QAAX,CvO4J1B,C;O;KuOzK1C,C;kFAeA,yB;MAAA,8C;MvOgJA,+B;MuOhJA,sC;QAa4D,OvO6IIB,euO7I2B,YvO8C9B,cuO9C8B,EAA Y,QAAX,CvO6I3B,C;O;KuO1J1C,C;oFAeA,yB;MrO0FI,6B;MqOrTJ,gD;MA2NA,4B;QAM+C,OA3NM,atOyShB,CCeE,cAAAU,cAAL,GAAiB,KAAtB,CDfF,MsOzSgB,C;O;KAqNrD,C;oGAQA,yB;MtHnLA,4B;MsHmLA,4B;QAMuD,OtHnLI,Y/G4NIB,c+NrdpC,GAAY,KhHvK0C,CgHuKvD,GAA+C,EAA/C,I;O;KMMJ,C;sGAQA,yB;MNZA,kE;M MYA,4B;QAMwD,ONZoB,sB/N6CnC,c+N7Ce,GAAW,KAAS,C;O;KMM5E,C;8FAQA,yB;MAAA,0D;MrOiA,4B;MAAA,iC;MqOjIA,4B;QAOuD,OrOqIZ,gBAA Y,QqOrIQ,kBrOkCrB,cAAL,GAAiB,KqO1CS,CrOqIR,CAAZ,C;O;KqO5I3C,C;4FASA,yB;MAAA,wD;MrOwHA,4B;MAAA,iC;MqOxHA,4B;QAOsD,OrO4HX,gBAA Y,QqO5H

O,iBrOyBpB,cAAL,GAAiB,KqOzBQ,CrO4HP,CAAZ,C;O;KqOnI3C,C;gFAUA,yB;MAAA,4C;MrOmGA,iC;Mq OnGA,sC;QAa6D,OrOgGhB,gBqOhG0B,WrOD9B,cqOC8B,EAAW,QAAX,CrOgG1B,C;O;KqO7G7C,C;kFAeA, yB;MAAA,8C;MrOoFA,iC;MqOpFA,sC;QAa8D,OrOiFjB,gBqOjF2B,YrOhB/B,cqOgB+B,EAAZ,QAAX,CrOiF3 B,C;O;KqO9F7C,C;ICtRA,qC;MAEI,SvOuIoD,cuOvI3C,CvOuI2C,EuOvIvC,CvOuIuC,C;MuOtIpD,SvOsIoD,cuO tl3C,CvOsI2C,EuOtIvC,CvOsIuC,C;MuOrIpD,OvOmDkE,YuOnDvD,EvOmDwE,KAAjB,EuOnDjD,EvOmD8E,K AA7B,CuOnDvD,KAAZ,GvOkFsD,SuOIFjC,EvOkF2C,KAAK,GuOIF3C,EvOkFuD,KAAZ,IAAf,CuOIFtD,GvOq EqD,SAAU,CAaT,SuOIFpB,EvOkF8B,KAAK,GuOIF9B,EvOkF0C,KAAZ,IAAf,CaBS,MAAK,GuOrExB,CvOqE mC,KAAZ,IAAf,C;K;IuOIEzD,qC;MACI,StNwIsD,esNxI7C,CtNwI6C,EsNxIzC,CtNwIyC,C;MsNvItD,StNuIsD,e sNvI7C,CtNuI6C,EsNvIzC,CtNuIyC,C;MsNtItD,OtNqDmE,asNrDxD,EtNqD0E,KAAiB,EsNrDID,EtNqDgF,KAA 9B,CsNrDxD,KAAZ,GtN+EwD,UsN/EnC,EtN+E8C,KAAK,UsN/E9C,EtN+E0D,KAAZ,CAAhB,CsN/ExD,GtNk EuD,UAAW,CAaV,UsN/EtB,EtN+EiC,KAAK,UsN/EjC,EtN+E6C,KAAZ,CAAhB,CaBU,MAAK,KsNIE3B,CtNk EsC,KAAZ,CAAhB,C;K;IsN/D3D,uD;MAmBI,WAAO,CAAP,C;QAD8E,OvOwBZ,YuOvBID,KvOuBmE,KAAjB ,EuOvBzC,GvOuBsE,KAA7B,CuOvBID,KAD8D,GACHD,GADgD,GvOuDxB,SuOtDf,GvOsDyB,KAAK,GuOtDx B,mBAAiB,GAAjB,EAAAsB,KAAtB,EvOqXV,SuOrXuC,IvOqXvC,CuOrXU,CvOsDoC,KAAZ,IAAf,C;auOrDtD, WAAO,CAAP,C;QAF8E,OvOwBZ,YuOtBID,KvOsBmE,KAAjB,EuOtBzC,GvOsBsE,KAA7B,CuOtBID,KAF8D, GAEhD,GAFgD,GvO0CzB,SuOxCd,GvOwCwB,KAAK,GuOxCvB,mBAAiB,KAAjB,EAAwB,GAAxB,EvOoXV, SuOpXwC,CAAC,IAAD,IvOoXxC,CuOpXU,CvOwCkC,KAAZ,IAAf,C;;QuOvC7C,MAAA,gCAAYB,eAAzB,C;K ;IAGzB,uD;MAmBI,sBAAO,CAAP,C;QADkF,OtNqF,asNPnD,KtNOqE,KAAiB,EsNP1C,GtNOwE,KAA9B,CsNP nD,KADkE,GACpD,GADoD,GtNkC1B,UsNjCjB,GtNiC4B,KAAK,UsNjC3B,mBAAiB,GAAjB,EAAAsB,KAAtB,E tN4WP,UsN5WoC,ItN4WpC,CsN5WO,CtNiCuC,KAAZ,CAAhB,C;asNhCxD,sBAAO,CAAP,C;QAFkF,OtNqF,as NNnD,KtNMqE,KAAiB,EsNN1C,GtNMwE,KAA9B,CsNNnD,KAFkE,GAEPD,GAFOd,GtNqB3B,UsNnBhB,GtN mB2B,KAAK,KsNnB1B,mBAAiB,KAAjB,EAAwB,GAAxB,EtN2WP,UsN3WsC,IAAD,atN2WrC,CsN3WO,CtN mBqC,KAAZ,CAAhB,C;;QsNIB/C,MAAA,gCAAYB,eAAzB,C;K;ItOIDC,sB;MAEtB,8B;MAFMg,gB;K;IAEnG,4 B;MAAA,gC;MACI,iBAGqC,WAAO,CAAP,C;MAErC,iBAGqC,WAAO,MAAP,C;MAErC,kBAGmC,C;MAEnC, iBAGkC,E;K;;IANbtC,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;wGAsBA,iB;MAM0D,OAAa,0BAuPjC,SAAL,G AAiB,KAvPqB,EAAU,KFsP3C,KAAL,GAAiB,GEtPqB,C;K;OGAEvE,iB;MAOoE,OAAa,0BA8O3C,SAAL,GAAi B,KA9O+B,EAAU,KA8OrD,KAAL,GAAiB,KA9O+B,C;K;wGAEjF,yB;MAqQA,6B;MDtQA,8C;MCCA,wB;QA MyD,ODAS,YAAiB,CCwQhD,cAAU,SAAL,GAAiB,KAAtB,CDxQgD,MAAJB,ECAe,KDAc,KAA7B,C;O;KCNi E,C;wGAQA,yB;MAuQA,aAS6D,0B;MAT7D,+B;MgBxQA,gD;MhBCA,wB;QAM0D,OgBAS,aAAkB,ChB0QhD ,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB1QgD,MAAIB,EhBAGB,KgBAc,KAA9B,C;O;KhBNnE,C;8FAGA,yB; MAqPA,6B;MArPA,wB;QAEsD,ODMD,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GAAW, CDqP5C,cE3PsC,KF2P5B,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KCRrD,C;8FAGA,yB;MAkPA,6B; MAIPA,wB;QAEuD,ODGF,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GAAW,CCsP5C,cAzP uC,KAYP7B,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KCLrD,C;8FAGA,yB;MA+OA,6B;MA/OA,wB;Q AEqD,ODAA,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GCAI,KDAO,KAAZ,IAAf,C;O;KC FrD,C;8FAGA,yB;MAsPA,aAS6D,0B;MAT7D,+B;MatPA,wB;QAEuD,OgBAA,eAAW,ChB6P7B,eAAW,oBAA L,SAAK,CAAL,YAAN,CgB7P6B,MAAK,KhBAI,KgBAO,KAAZ,CAAhB,C;O;KhBFvD,C;gGAIA,yB;MAwOA, 6B;MAxOA,wB;QAEuD,ODMD,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GAAZ,CDwO9 C,cE9OwC,KF8O9B,KAAL,GAAiB,GAAtB,CCxO8C,MAAZ,IAAf,C;O;KCRtD,C;gGAGA,yB;MAqOA,6B;MAr OA,wB;QAEwD,ODGF,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GAAZ,CCyO9C,cA5Oy C,KA4O/B,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KCLtD,C;gGAGA,yB;MAkOA,6B;MAIOA,wB;Q AEsD,ODAA,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GCAK,KDAO,KAAZ,IAAf,C;O;K CFtD,C;gGAGA,yB;MAyOA,aAS6D,0B;MAT7D,+B;MAzOA,wB;QAEwD,OgBAA,eAAW,ChBgP9B,eAAW,oB AAL,SAAK,CAAL,YAAN,CgBhP8B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O;KhBFxD,C;gGAIA,yB;MA2 NA,6B;MA3NA,wB;QAEuD,ODMD,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,EA AY,CD2N9C,cEjOwC,KFiO9B,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;KCRtD,C;gGAGA,yB;MAw NA,6B;MAxNA,wB;QAEwD,ODGF,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,EAA Y,CC4N9C,cA/NyC,KA+N/B,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KCLtD,C;gGAGA,yB;MAqN

A,6B;MArNA,wB;QAEsD,ODAA,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,ECAK, KDAO,KAAZ,CAAf,C;O;KCFtD,C;gGAGA,yB;MA4NA,aAS6D,0B;MAT7D,+B;MA5NA,wB;QAEwD,OgBAA, eAAW,ChBm09B,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBn08B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O ;KhBFxD,C;4FAIA,yB;MA8MA,6B;MDxMA,4C;MCNA,wB;QAEqD,ODMD,WC+MjB,cAAU,SAAL,GAAiB,K AAtB,CD/MiB,ED8MjB,cEpNoC,KFoN1B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KCRpD,C;4FAGA,yB;MA2MA ,6B;MDxMA,4C;MCHA,wB;QAEsD,ODGF,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,EC+MjB,cAlNqC, KAKN3B,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KCLpD,C;4FAGA,yB;MAwMA,6B;MDxMA,4C;MCAA,wB;QA EoD,ODAA,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,ECakB,KDAIB,C;O;KCFpD,C;4FAGA,yB;MA+M A,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MhBAA,wB;QAEsD,OgBAA,YhBsNjB,eAAW,oBAAL,SAAK,CAAL,Y AAN,CgBtNiB,EhBAmB,KgBAnB,C;O;KhBFtD,C;4FAIA,yB;MAiMA,6B;MDnLA,kD;MCdA,wB;QAMqD,ODc D,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,EDqLjB,cEnMoC,KFmM1B,KAAL,GAAiB,GAAtB,CCrLiB,C; O;KCpBpD,C;4FAOA,yB;MA0LA,6B;MDnLA,kD;MCPA,wB;QAMsD,ODOF,cCsLjB,cAAU,SAAL,GAAiB,KA AtB,CDtLiB,ECsLjB,cA7LqC,KA6L3B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KCbpD,C;4FAOA,yB;MAMLA,6B; MDnLA,kD;MCAA,wB;QAMoD,ODAA,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECakB,KDAIB,C;O;KC NpD,C;4FAOA,yB;MAsLA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MhBAA,wB;QAMsD,OgBAA,ehByLjB,eAAW, oBAAL,SAAK,CAAL,YAAN,CgBzLiB,EhBAmB,KgBAnB,C;O;KhBNtD,C;sGAQA,yB;MAoKA,6B;MDxMA,4 C;MCoCA,wB;QAMiD,ODxCG,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,ED8MjB,cEtKqC,KFsK3B,KA AL,GAAiB,GAAtB,CC9MiB,C;O;KCKCpD,C;sGAOA,yB;MA6JA,6B;MDxMA,4C;MC2CA,wB;QAMkD,OD/CE ,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,EC+MjB,cAhKsC,KAgK5B,KAAL,GAAiB,KAAtB,CD/MiB,C ;O;KCYCpD,C;sGAOA,yB;MAsJA,6B;MDxMA,4C;MCKDA,wB;QAMgD,ODtDI,WC+MjB,cAAU,SAAL,GAAiB ,KAAtB,CD/MiB,ECsDmB,KDtDnB,C;O;KCGDpD,C;sGAOA,yB;MAyJA,aAS6D,0B;MAT7D,+B;MgB/MA,8C; MhBsDA,wB;QAMkD,OgB1DI,YhBsNjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBtNiB,EhB0DoB,KgB1DpB,C ;O;KhBoDtD,C;4FAQA,yB;MAuIA,6B;MDnLA,kD;MDiPJ,0B;MAAA,+B;MErMI,wB;QAQ6C,OFwMR,eAAW, OCtPI,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,EDqLjB,cEvI4B,KFuIIB,KAAL,GAAiB,GAAtB,CCrLiB,C A4Lf,KD0DW,CAAX,C;O;KEhNrC,C;4FASA,yB;MA8HA,6B;MDnLA,kD;MCKPJ,4B;MAAA,iC;MA7LI,wB;Q AQ+C,OAgMR,gBAAY,QDvPC,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECsLjB,cA/H8B,KA+HpB,KAA L,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C;O;KAXMvC,C;4FASA,yB;MAqHA,6B;MDnLA,kD;MC8DA ,wB;QAQ2C,ODhES,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECgES,KDhET,C;O;KCwDpD,C;4FASA,yB; MAsHA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MhBgEA,wB;QAQ6C,OgBIES,ehByLjB,eAAW,oBAAL,SAAK,CA AL,YAAN,CgBzLiB,EhBkEU,KgBIEV,C;O;KhB0DtD,C;4EAUA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C, uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;4EAQA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,Q AAL,SAAK,KAAZ,C;O;KAN3C,C;oGAQA,yB;MAAA,8C;MAkFA,6B;MAIFA,wB;QAE+D,0BAyF5B,cAAU,S AAL,GAAiB,KAAtB,CAzF4B,EAYF5B,cAzFqD,KAYF3C,KAAL,GAAiB,KAAtB,CAzF4B,C;O;KAF/D,C;0GAI A,yB;MAAA,yC;MA8EA,6B;MA9EA,wB;QAQKE,aA+E/B,cAAU,SAAL,GAAiB,KAAtB,CA/E+B,EA+E/B,cA/E mD,KA+EzC,KAAL,GAAiB,KAAtB,CA/E+B,C;O;KARIE,C;4FAUA,yB;MAAA,iC;M4L7NJ,4B;M5L6NI,wB;Q AEqD,uB4L5NiC,Q5L4N1B,IAAK,K4L5NX,G5L4NoB,KAAM,K4L5NM,C5L4NjC,C;O;KAFrD,C;0FAGA,yB; MAAA,iC;M4L3NJ,4B;M5L2NI,wB;QAEoD,uB4L1NgC,Q5L0NzB,IAAK,K4L1NX,G5L0NmB,KAAM,K4L1N M,C5L0NhC,C;O;KAFpD,C;4FAGA,yB;MAAA,iC;M4LzNJ,4B;M5LyNI,wB;QAEqD,uB4LxNiC,Q5LwN1B,IAA K,K4LxNX,G5LwNoB,KAAM,K4LxNM,C5LwNjC,C;O;KAFrD,C;4EAGA,yB;MAAA,iC;M4LvNJ,4B;M5LuNI, mB;QAEkC,uB4LtnsB,QAAP,C5LsNR,S4LtnE,C5LsntB,C;O;KAFIC,C;kFAIA,yB;MAAA,0B;MAAA,mB;QAU mC,OAAK,OAAL,SAAK,C;O;KAVxC,C;oFAWA,Y;MASqC,gB;K;gFACrC,Y;MASiC,OAAK,SAAL,GAAiB,K; K;kFACID,yB;MAAA,aASqD,0B;MATrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,Y;O;KATnC,C;oFAWA,y B;MF+DJ,0B;MAAA,+B;ME/DI,mB;QASqC,OFiEE,eAAW,OEjEb,SfiEa,CAAX,C;O;KE1EvC,C;sFAUA,Y;MA EuC,W;K;kFACvC,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,KAAtB,C;O;KATnC,C;oFAUA,y B;MAAA,aAS6D,0B;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,YAAN,C;O;KATrC,C;oFA WA,Y;MAMqC,OApDC,SAAL,GAAiB,K;K;sFAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,K;K;gCA6DID,Y;MA AyC,OAAQ,CA7DX,SAAL,GAAiB,KA6DD,Y;K;,,,,gCArVrD,Y;MAAA,c;MAGuG,qD;MAHvG,a;K;8BAAA,iB; MAAA,2IAGuG,oCAHvG,G;K;0EAyVA,yB;MAAA,iC;MAAA,4B;QAW4C,uBAAY,SAAZ,C;O;KAX5C,C;4EA

YA,yB;MAAA,iC;MAAA,4B;QAU6C,uBAAO,SAAP,C;O;KAV7C,C;4EAWA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW2C,uBAAY,QAAL,SAAK,CAAZ,C;O;KAX3C,C;4EAYA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW4C,uBAAY,QAAL,SAAK,SAAZ,C;O;KAX5C,C;IkCzXA,8B;MACqB,sB;K;wCAKjB,iB;MAM8C,OICgWL,WkChWK,aAAQ,KAAR,CICgWL,C;K;wCkC9VzC,wB;MAOI,aAAQ,KAAR,IAAiB,KICsPgB,K;K;mFkCIPP,Y;MAAQ,OAAA,YAAQ,O;K;qCAE9C,Y;MAC+E,gCAAS,YAAT,C;K;IAEzD,qC;MAAC,oB;MACnB,eAAoB,C;K;6CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;0CACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OIC2Ue,WkC3US,aAAM,mBAAN,EAAM,2BAAN,OIC2UT,C;;QkC3UwC,MAAM,2BAAuB,YAAM,WAA7B,C;K;;2CAGvF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,K;MAEzC,OAAe,WAAR,YAAQ,EAAS,OICiOS,KkCjOIB,C;K;gDAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QjBioDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UiBjoD6B,2BjBioDR,OiBjoDQ,S;UAAA,W;YAAwB,oBAAR,YAAQ,EjBioDhC,OjBp6CA,KkC7NgC,C;;UjBioD/C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MiBloDH,iB;K;oCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA9CvD,uC;MAAA,qD;MACgC,wBAAK,eAAW,IAAX,CAAL,C;MADhC,Y;K;;;qCAPJ,Y;MAAAA,OAKqB,sDALrB,M;K;qCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;mCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;kFAwDA,yB;MAAA,2C;MAWwC,0C;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CIC4MV,K;S;O;MkCvNzC,6B;QAWI,OAAO,qBAAY,gCAAW,IAAX,GAAiB,wBAAjB,CAAZ,C;O;KAXX,C;oFAcA,oB;MAGwE,e;K;IqM3ExE,sC;MAQ2D,OAAa,WAAb,SzOkRjB,KAAL,GAAiB,GyOIRkB,EAAS,KAAT,C;K;IAExE,sC;MAQ4D,OAAa,WAAb,SvOyQIB,KAAL,GAAiB,KuOzQmB,EAAS,KAAT,C;K;IAGzE,sC;MAQ0D,OAAc,WxO2R5B,oBwO3Rc,SxO2RnB,KAAK,CAAL,iBwO3RiC,EAAS,KAAT,C;K;IAExE,sC;MAOgD,uBAAc,SvNmRvB,KuNnRS,EAA6B,WAAW,KAAX,CAA7B,C;K;IAGhD,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAGvE,+B;MAMuC,Q;MAAA,2DAAoB,kBAAkB,SAAlB,C;K;IAE3D,sC;MAOiD,Q;MAAA,2CAAE,KAAf,oBAAYB,kBAAkB,SAAlB,C;K;IAE1E,6B;MAMmC,Q;MAAA,yDAaB,kBAAkB,SAAlB,C;K;IAErD,oC;MAO6C,Q;MAAA,yCAAa,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAEpE,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAMvE,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MAS0C,IAAvB,I;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,IxO/EkE,YwO+E9D,GxO/E+E,KAAjB,EAA6B,CDuQ5D,SyOxLzB,6BAAM,UzOwL6B,KAAL,GAAiB,GAAtB,CCvQ4D,MAA7B,CwO+E9D,IAAJ,C;QAA2B,OAAO,I;MACIC,OzOwPqC,UAAW,OyOxPzC,GxO8L8B,KD0DW,CAAX,C;K;IyOrPzC,mC;MAM8C,mCAAuB,EAAvB,C;K;IAE9C,4C;MAS0C,IAAvB,I;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,IxOrGkE,YwOqG9D,GxOrG+E,KAAjB,EAA6B,CCwQ5D,SuOnKzB,8BAAO,UvOmK4B,KAAL,GAAiB,KAAtB,CDxQ4D,MAA7B,CwOqG9D,IAAJ,C;QAA4B,OAAO,I;MACnB,OvOmOuB,WAAy,QuOnO5C,GxOkLgC,KCiDY,CAAZ,C;K;IuOhO3C,iC;MAM0C,iCAAqB,EAArB,C;K;IAE1C,0C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAkB,4BAAK,U;MACvB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MACHB,IAAI,YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,QAAQ,C;;MAGZ,uBAAuB,mB;MAEvB,qBAaqB,gB;MACrB,axOiNmC,SwOjNtB,KxOiNsB,C;MwOhNnC,aAAa,W;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CAAR,EAaiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IxOnJ8D,YwOmJ1D,MxOnJ2E,KAAjB,EwOmJjD,cxOnJ8E,KAA7B,CwOmJ1D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBxO5FwC,WwO4FvB,KxO5FuB,EwO4Ff,MxO5Fe,C;YwO8FxC,IxOvJsd,YwOuJID,MxOvJmE,KAAjB,EwOuJzC,cxOvJsE,KAA7B,CwOuJID,IAAJ,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIIf,SxOnHkD,SAAE,YwOmHjE,MxOnH4D,KAAK,EwOmHvD,MxOnHmE,KAAZ,CAAF,C;QwOqHID,mBAAmB,M;QACnB,SxOhJiD,SwOgJjD,MxOhJ2D,KAAK,GAAW,CA4U5C,SwO5LrB,KxO4LqB,CA5U4C,MAAX,IAAf,C;QwOiJjD,IxOnK8D,YwOmK1D,MxOnK2E,KAAjB,EwOmKjD,YxOnK8E,KAA7B,CwOmK1D,IAAJ,C;UAA2B,OAAO,I;;MAGtC,OAAO,M;K;IAGX,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAmB,6BAAM,U;MACzB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MACHB,IAAI,YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,QAAQ,C;;MAIZ,uBAAuB,gD;MAEvB,qBAaqB,gB;MACrB,avNoJqC,UAAW,oBuNpJnC,KvNoJmC,CAAX,C;MuNnJrC,aAAa,2B;MACb,aAAU,KAAV,MAAsB,MAAtB,







// this software without specific prior written permission.

Found in path(s):

```
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/type.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/any.proto
*
/opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/struct.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/empty.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-
jar/google/protobuf/source_context.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/duration.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/1720055189_1686654514.924605/0/protobuf-java-3-22-5-jar/google/protobuf/api.proto
```

## 1.193 snake-yaml 2.0

### 1.193.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// This module is multi-licensed and may be used under the terms
// EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal
// LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html
// GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html
// AL, Apache License, V2.0 or later, http://www.apache.org/licenses
// BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php
/**
 * A Base64 encoder/decoder.
 *
 * <p>
 * This class is used to encode and decode data in Base64 format as described in RFC 1521.
 *
 * <p>
 * Project home page: www.
 * source-code.biz/base64coder/java

 * Author: Christian d'Heureuse, Inventec Informatik AG, Zurich, Switzerland

 * Multi-licensed: EPL / LGPL / GPL / AL / BSD.
 */
```

Found in path(s):

```
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/external/biz/base64Coder/Base64Coder.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

\* Copyright (c) 2008 Google Inc.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/Escaper.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/PercentEscaper.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/UnicodeEscaper.java  
No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>  
<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/META-INF/maven/org.yaml/snakeyaml/pom.xml  
No license file was found, but licenses were detected in source scan.

/\*\*

\* Copyright (c) 2008, SnakeYAML  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-

jar/org/yaml/snakeyaml/util/ArrayStack.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/tokens/StreamStartToken.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/introspector/PropertyUtils.java  
 \*  
 /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/resolver/Resolver.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/nodes/AnchorNode.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/tokens/BlockEntryToken.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/util/EnumUtils.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/constructor/AbstractConstruct.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/constructor/SafeConstructor.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/TypeDescription.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/scanner/Constant.java  
 \*  
 /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/tokens/AliasToken.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/Yaml.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/constructor/CustomClassLoaderConstructor.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/emitter/EmitterException.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/events/NodeEvent.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/composer/ComposerException.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/resolver/ResolverTuple.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/tokens/DocumentStartToken.java  
 \*  
 /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/error/MarkedYAMLEException.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/util/PlatformFeatureDetector.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/events/SequenceStartEvent.java  
 \* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
 jar/org/yaml/snakeyaml/tokens/KeyToken.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/MappingEndEvent.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/extensions/compactnotation/PackageCompactConstructor.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/reader/StreamReader.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/comments/CommentEventsCollector.java

\*

/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/ScalarNode.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/scanner/ScannerImpl.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/TagTuple.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/error/YAMLException.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/StreamStartEvent.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/CollectionNode.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/emitter/ScalarAnalysis.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/inspector/TrustedTagInspector.java

\*

/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/Event.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/inspector/TrustedPrefixesTagInspector.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/Node.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/emitter/Emitable.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/FlowSequenceStartToken.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/scanner/Scanner.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/DocumentEndEvent.java

\*

/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/representer/Representer.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/BlockMappingStartToken.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/parser/ParserImpl.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-

jar/org/yaml/snakeyaml/events/CollectionEndEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/scanner/ScannerException.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/MissingProperty.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/internal/Logger.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/parser/ParserException.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/tokens/StreamEndToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/MethodProperty.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/tokens/BlockEndToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/util/ArrayUtils.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/events/ScalarEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/parser/Production.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/comments/CommentType.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/serializer/Serializer.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/extensions/compactnotation/CompactData.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/nodes/NodeId.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/env/EnvScalarConstructor.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/FieldProperty.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/Property.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/tokens/DirectiveToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/error/MissingEnvironmentVariableException.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/BeanAccess.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/introspector/GenericProperty.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-  
jar/org/yaml/snakeyaml/emitter/Emitter.java

\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/representer/SafeRepresenter.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/serializer/SerializerException.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/FlowMappingEndToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/BlockSequenceStartToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/DocumentStartEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/inspector/TagInspector.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/AnchorToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/LoaderOptions.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/CommentEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/AliasEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/serializer/NumberAnchorGenerator.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/CommentToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/constructor/ConstructorException.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/comments/CommentLine.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/Tag.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/DumperOptions.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/FlowEntryToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/representer/BaseRepresenter.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/composer/Composer.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/ValueToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/MappingNode.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/reader/ReaderException.java  
\*

/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/DocumentEndToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/Token.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/SequenceEndEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/NodeTuple.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/serializer/AnchorGenerator.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/constructor/DuplicateKeyException.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/CollectionStartEvent.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/TagToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/ScalarToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/parser/VersionTagsTuple.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/ImplicitTuple.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/FlowMappingStartToken.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/events/MappingStartEvent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/extensions/compactnotation/CompactConstructor.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/tokens/FlowSequenceEndToken.java  
\*  
/opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/constructor/Constructor.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/util/UriEncoder.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/representer/Represent.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/error/Mark.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/nodes/SequenceNode.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/inspector/UnTrustedTagInspector.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/introspector/PropertySubstitute.java  
\* /opt/cola/permits/1720484947\_1686744817.7895513/0/snakeyaml-2-0-sources-6-jar/org/yaml/snakeyaml/constructor/BaseConstructor.java

```

*
/opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/scanner/SimpleKey.java
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/constructor/Construct.java
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/emitter/EmitterState.java
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/reader/UnicodeReader.java
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/events/StreamEndEvent.java
* /opt/cola/permits/1720484947_1686744817.7895513/0/snakeyaml-2-0-sources-6-
jar/org/yaml/snakeyaml/parser/Parser.java

```

# 1.194 openjdk-jre 17.0.7u7

## 1.194.1 Available under license :

```
c-libutl 20160225
```

```
c-libutl License
```

```
...
```

This software is distributed under the terms of the BSD license.

```
== BSD LICENSE =====
```

(C) 2009 by Remo Dentato (rdentato@gmail.com)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT



(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<http://opensource.org/licenses/bsd-license.php>

...

## Apache Xerces v2.12.2

### Apache Xerces Notice

<pre>

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
=====
```

Apache Xerces Java  
Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

</pre>

### Apache 2.0 License

<pre>

```
 Apache License
 Version 2.0, January 2004
 http://www.apache.org/licenses/
```

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean  
an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with

the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any

Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of

TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

</pre>

## JLine v3.20.0

### JLine License

<pre>

Copyright (c) 2002-2018, the original author or authors.  
All rights reserved.

<https://opensource.org/licenses/BSD-3-Clause>

Redistribution and use in source and binary forms, with or  
without modification, are permitted provided that the following  
conditions are met:

Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer  
in the documentation and/or other materials provided with  
the distribution.

Neither the name of JLine nor the names of its contributors  
may be used to endorse or promote products derived from this  
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,  
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY  
AND

FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO  
EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,  
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED  
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING  
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

Copyright (c) 1999-2003 David Corcoran <corcoran@musclecard.com>

Copyright (c) 2001-2011 Ludovic Rousseau <ludovic.rousseau@free.fr>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN

NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some files are under GNU GPL v3 or any later version

- doc/example/pcsc\_demo.c

- the files in src/spy/

- the files in UnitaryTests/

Copyright (C) 2003-2014 Ludovic Rousseau

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Files src/auth.c and src/auth.h are:

\* Copyright (C) 2013 Red Hat

\*

\* All rights reserved.

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

\* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

\* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS

\* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

\* COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,

\* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS

\* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

\* AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

\* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF

\* THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH

\* DAMAGE.

\*

\* Author: Nikos Mavrogiannopoulos <nnav@redhat.com>

Files src/simclist.c and src/simclist.h are:

\* Copyright (c) 2007,2008,2009,2010,2011 Mij <mij@bitchx.it>

\*

\* Permission to use, copy, modify, and distribute this software for any

\* purpose with or without fee is hereby

\* granted, provided that the above

\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES

\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

\* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN

\* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF

\* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.



```
xwd v1.0.7
```

```
xwd utility
```

```
<pre>
```

This is the copyright for the files in src/java.desktop/unix/native/libawt\_xawt:  
list.h, multiVis.h, wsutils.h, list.c, multiVis.c

Copyright 1994 Hewlett-Packard Co.

Copyright 1996, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

```
</pre>
```

```
IAIK (Institute for Applied Information Processing and Communication) PKCS#11 wrapper files v1
```

```
IAIK License
```

```
<pre>
```

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of Technology."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

## The FreeType Project: Freetype v2.12.1

### FreeType Notice

...

FreeType comes with two licenses from which you can choose the one which fits your needs best.

The FreeType License (FTL) is the most commonly used one. It is a BSD-style license with a credit clause and thus compatible with the GNU Public License (GPL) version 3, but not with the GPL version 2.

The GNU General Public License (GPL), version 2. Use it for all projects which use the GPLv2 also, or which need a license compatible to the GPLv2.

...

### FreeType License

...

Copyright (C) 1996-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2007-2022 by Dereg Clegg and Michael Toftdal.  
Copyright (C) 1996-2022 by Just van Rossum, David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2004-2022 by Masatake YAMATO and Redhat K.K.  
Copyright (C) 2007-2022 by Derek Clegg and Michael Toftdal.  
Copyright (C) 2007-2022 by David Turner.  
Copyright (C) 2022 by David Turner,  
Robert Wilhelm, Werner Lemberg, and Moazin Khatti.  
Copyright (C) 2007-2022 by Rahul Bhalerao <rahul.bhalerao@redhat.com>, <b.rahul.pm@gmail.com>.  
Copyright (C) 2008-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and suzuki toshiya.  
Copyright (C) 2019-2022 by Nikhil Ramakrishnan, David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright (C) 2009-2022 by Oran Agra and Mickey Gabel.  
Copyright (C) 2004-2022 by David Turner, Robert Wilhelm, Werner Lemberg, and George Williams.  
Copyright (C) 2004-2022 by Masatake YAMATO, Red Hat K.K.,  
Copyright (C) 2003-2022 by Masatake YAMATO, Redhat K.K.,  
Copyright (C) 2013-2022 by Google, Inc.  
Copyright (C) 2018-2022 by David Turner, Robert Wilhelm, Dominik Rtsches, and Werner Lemberg.  
Copyright (C) 2005-2022 by David Turner, Robert Wilhelm, and Werner Lemberg.  
Copyright 2013 by Google, Inc.

## The FreeType Project LICENSE

-----

2006-Jan-27

Copyright 1996-2002, 2006 by  
David Turner, Robert Wilhelm, and Werner Lemberg

## Introduction

=====

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the

## FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. (^ as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. (^ royalty-free' usage)
- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (^ credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

""""

Portions of this software are copyright <year> The FreeType Project ([www.freetype.org](http://www.freetype.org)). All rights reserved.

""""

Please replace <year> with the value from the FreeType version you actually use.

## Legal Terms

=====

### 0. Definitions

-----

Throughout this license, the terms `package`, `FreeType Project`, and `FreeType archive` refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project`, be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive.

If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

#### 1. No Warranty

-----

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

#### 2. Redistribution

-----

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file

(`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

-----

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

### 4. Contacts

-----

There are two mailing lists related to FreeType:

- o [freetype@nongnu.org](mailto:freetype@nongnu.org)

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution.

If you are looking for support, start in this list if you

haven't found anything to help you in the documentation.

o freetype-devel@nongnu.org

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

<http://www.freetype.org>

...

### GPL v2

...

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it.

By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally,  
any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).



Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each

and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

...

```
Additional Freetype Attributions
...
```

-----

The below license applies to the following files:

```
libfreetype/src/psaux/psarrst.c
libfreetype/src/psaux/psarrst.h
libfreetype/src/psaux/psblues.c
```

libfreetype/src/psaux/psblues.h  
libfreetype/src/psaux/pserror.c  
libfreetype/src/psaux/pserror.h  
libfreetype/src/psaux/psfixed.h  
libfreetype/src/psaux/psfont.c  
libfreetype/src/psaux/psfont.h  
libfreetype/src/psaux/psft.c  
libfreetype/src/psaux/psft.h  
libfreetype/src/psaux/psglue.h  
libfreetype/src/psaux/pshints.c  
libfreetype/src/psaux/pshints.h  
libfreetype/src/psaux/psintrap.c  
libfreetype/src/psaux/psintrap.h  
libfreetype/src/psaux/psread.c  
libfreetype/src/psaux/psread.h  
libfreetype/src/psaux/psstack.c  
libfreetype/src/psaux/psstack.h  
libfreetype/src/psaux/pstypes.h

#### Copyright

2006-2014 Adobe Systems Incorporated.

This software, and all works of authorship, whether in source or object code form as indicated by the copyright notice(s) included herein (collectively, the "Work") is made available, and may only be used, modified, and distributed under the FreeType Project License, LICENSE.TXT. Additionally, subject to the terms and conditions of the FreeType Project License, each contributor to the Work hereby grants to any individual or legal entity exercising permissions granted by the FreeType Project License and this section (hereafter, "You" or "Your") a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such contributor that are necessarily infringed by their contribution(s) alone or by combination of their contribution(s) with the Work to which such contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

By using, modifying, or distributing the Work you indicate that you have read and understood the terms and conditions of the FreeType Project License as well as those provided in this section,

and you accept  
them fully.

...

### MIT License  
...

-----  
The below license applies to the following files:  
libfreetype/include/freetype/internal/fthash.h  
libfreetype/src/base/fthash.c

Copyright 2000 Computing Research Labs, New Mexico State University  
Copyright 2001-2015

Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

...

## Harfbuzz v4.4.1

### Harfbuzz License

<https://github.com/harfbuzz/harfbuzz/blob/4.4.1/COPYING>

<pre>



HarfBuzz is licensed under the so-called "Old MIT" license. Details follow.  
For parts of HarfBuzz that are licensed under different licenses see individual  
files names COPYING in subdirectories where applicable.

Copyright 2010-2022 Google, Inc.  
Copyright 2018-2020 Ebrahim Byagowi  
Copyright 2019-2020 Facebook, Inc.  
Copyright 2012-2015 Mozilla Foundation.  
Copyright 2011 Codethink Limited  
Copyright 2008-2010 Nokia Corporation and/or its subsidiary(-ies)  
Copyright 2009 Keith Stribley  
Copyright 2009 Martin Hosken and SIL International  
Copyright 2007 Chris Wilson  
Copyright 2005-2022 Behdad Esfahbod  
Copyright 2005 David Turner  
Copyright 2004-2013 Red Hat, Inc.  
Copyright 1998-2004 David Turner and Werner Lemberg  
Copyright 2016 Elie Roux <elie.roux@telecom-bretagne.eu>  
Copyright 2018-2019 Adobe Inc.  
Copyright 2018 Khaled Hosny  
Copyright 2016 Igalia S.L.

For  
full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without  
license or royalty fees, to use, copy, modify, and distribute this  
software and its documentation for any purpose, provided that the  
above copyright notice and the following two paragraphs appear in  
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR  
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES  
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN  
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH  
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,  
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND  
FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS  
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO  
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

All source  
code, except for one section, is licensed as above. The one  
exception is licensed with a slightly different MIT variant:  
The contents of this directory are licensed under the following terms:

-----  
The below license applies to the following files:  
libharfbuzz/hb-ucd.cc

Copyright (C) 2012 Grigori Goronzy <greg@kinoho.net>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

</pre>

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## PC/SC Lite v1.9.5

### PC/SC Lite License

<pre>

The main parts of the code are using the BSD-like licence below:

Copyright (c) 1999-2003 David Corcoran <corcoran@linuxnet.com>  
Copyright (c) 2001-2011 Ludovic Rousseau <ludovic.rousseau@free.fr>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Changes to this license can be made only by the copyright author with explicit written consent.

#### THIS

SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some source code files are using other licences. See the [COPYING](<https://salsa.debian.org/rousseau/PCSC/-/blob/master/COPYING>) file for details.

</pre>

```
The Unicode Standard, Unicode Character Database, Version 13.0.0
```

```
Unicode Character Database
```

```
````
```

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s
Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement.
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S
DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE
TERMS AND CONDITIONS OF THIS AGREEMENT.
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE
THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2020 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the

"Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

...

jQuery UI v1.12.1

jQuery UI License

...

Copyright jQuery Foundation and other contributors, <https://jquery.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/jquery-ui>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code contained within the demos directory.

CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

...

This is the copyright file

```
## zlib v1.2.13
```

```
### zlib License
```

```
<pre>
```

Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

```
</pre>  
jpackage test license file (just some sample text).  
## Eastman Kodak Company: Portions of color management and imaging software
```

```
### Eastman Kodak Notice  
<pre>  
Portions Copyright Eastman Kodak Company 1991-2003  
</pre>  
## Thai Dictionary
```

```
### Thai Dictionary License  
<pre>
```

Copyright (C) 1982 The Royal Institute, Thai Royal Government.

Copyright (C) 1998 National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Ministry of Science Technology and Environment,
Thai Royal Government.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

Apache Santuario v2.3.0

Apache Santuario Notice

<pre>

Apache Santuario - XML Security for Java
Copyright 1999-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

It was originally based on software copyright (c) 2001, Institute for
Data Communications Systems, <<http://www.nue.et-inf.uni-siegen.de/>>.

The development of this software was partly funded by the European
Commission in the <WebSig> project in the ISIS Programme.

</pre>

Apache 2.0 License

<pre>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity
authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except
in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

</pre>

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files: *

Copyright: APPLICATION_COPYRIGHT

License: APPLICATION_LICENSE_TEXT

ADDITIONAL INFORMATION ABOUT LICENSING

Certain files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception.

Note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0 and may include FreeType. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result.

Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Failing to distribute notices associated with some files may also create unexpected legal consequences.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications

to this package which may subsequently be redistributed and/or involve the use of third party software.
Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.

This code is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 only, as published by the Free Software Foundation. Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License version 2 for more details (a copy is included in the LICENSE file that accompanied this code).

You should have received a copy of the GNU General Public License version 2 along with this work; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

Please contact
Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
or visit www.oracle.com if you need additional information or have any questions.

OASIS PKCS #11 Cryptographic Token Interface v3.0

OASIS PKCS #11 Cryptographic Token Interface License
<pre>

Copyright OASIS Open 2020. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website: [<http://www.oasis-open.org/policies-guidelines/ipr>]

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any

document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set

forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of

intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

</pre>

DOM Level 3 Core Specification v1.0

W3C Software Notice

<pre>

Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

</pre>

W3C License

<pre>

W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

- 1.The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- 2.Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

3. Notice of any changes or modifications to the files, including the date changes were made.

(We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This

version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

</pre>

```
## Cryptix AES v3.2.0
```

```
### Cryptix General License
```

```
<pre>
```

Cryptix General License

Copyright (c) 1995-2005 The Cryptix Foundation Limited.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are

met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
</pre>  
## libpng v1.6.38
```

```
### libpng License  
<pre>
```

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE
=====

PNG Reference Library License version 2

Copyright (c) 1995-2022 The PNG Reference Library Authors.
Copyright (c) 2018-2022 Cosmin Truta
Copyright (c) 1998-2018 Glenn Randers-Pehrson
Copyright (c) 1996-1997 Andreas Dilger
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising

from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabuddhe
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is

with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000

Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of

merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

TRADEMARK

=====

The name "libpng" has not been registered by the Copyright owners as a trademark in any jurisdiction. However, because libpng has been distributed and maintained world-wide, continually since 1995, the Copyright owners claim "common-law trademark protection" in any jurisdiction where common-law trademark is recognized.

</pre>

AUTHORS File Information

...

PNG REFERENCE LIBRARY AUTHORS

=====

This is the list of PNG Reference Library ("libpng") Contributing Authors, for copyright and licensing purposes.

- * Andreas Dilger
- * Cosmin Truta
- * Dave Martindale
- * Eric S. Raymond

- * Gilles Vollant
- * Glenn Randers-Pehrson
- * Greg Roelofs
- * Guy Eric Schalnat
- * James Yu
- * John Bowler
- * Kevin Bracey
- * Magnus Holmgren
- * Mandar Sahastrabudde
- * Mans Rullgard
- * Matt Sarett
- * Mike Klein
- * Pascal Massimino
- * Paul Schmidt
- * Qiang Zhou
- * Sam Bushell
- * Samuel Williams
- * Simon-Pierre Cadieux
- * Tim Wegner
- * Tom Lane
- * Tom Tanner
- * Vadim Barkov
- * Willem van Schaik
- * Zhijie Liang
- * Arm Holdings
 - Richard Townsend
- * Google Inc.
 - Matt Sarett
 - Mike Klein
 - Dan Field
 - Sami Boukortt

The build projects, the build scripts, the test scripts, and other files in the "ci", "projects", "scripts" and "tests" directories, have other copyright owners, but are released under the libpng license.

Some files in the "contrib" directory, and some tools-generated files that are distributed with libpng, have other copyright owners, and are released under other open source licenses.

...

jopt-simple v5.0.4

MIT License

<pre>

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

jQuery v3.6.1

jQuery License

...

jQuery v 3.6.1

Copyright OpenJS Foundation and other contributors, <https://openjsf.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The jQuery JavaScript Library v3.6.1 also includes Sizzle.js

Sizzle.js includes the following license:

Copyright JS Foundation and other contributors, <https://js.foundation/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/sizzle>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

...

Apache Xalan v2.7.2

Apache Xalan Notice

<pre>

```
=====
====
== NOTICE file corresponding to the section 4d of the Apache License, Version 2.0, ==
== in this case for the Apache Xalan distribution.          ==
=====
====
```

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

Specifically, we only include the XSLTC portion of the source from the Xalan distribution.
The Xalan project has two processors: an interpretive one (Xalan Interpretive) and a
compiled one (The XSLT Compiler (XSLTC)). We *only* use the XSLTC part of Xalan; We use
the source from the packages that are part of the XSLTC sources.

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation.,
<http://www.lotus.com>.
- software copyright (c) 2001-2002, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation., <http://www.ibm.com>.
- voluntary contributions made by Ovidiu Predescu (ovidiu@cup.hp.com) on behalf of the
Apache Software Foundation and was originally developed at Hewlett Packard Company.

</pre>

Apache 2.0 License

<pre>

```
        Apache License
        Version 2.0, January 2004
        http://www.apache.org/licenses/
```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control,

are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare
Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative

Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to

in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian
Permission to use, copy, modify, and distribute this software and
its documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both the copyright notice and this permission notice and warranty
disclaimer appear in supporting documentation, and that the name of
the authors or their employers not be used in advertising or publicity
pertaining to distribution of the software without specific, written
prior permission.

The authors and their employers disclaim all warranties with regard to
this software, including all implied warranties of merchantability
and

fitness. In no event shall the authors or their employers be liable for
any special, indirect or consequential damages or any damages whatsoever
resulting from loss of use, data or profits, whether in an action of
contract, negligence or other tortious action, arising out of or in
connection with the use or performance of this software. The portions of
JLex output which are hard-coded into the JLex source code are (naturally)
covered by this same license.

</pre>

Little Color Management System (LCMS) v2.14

LCMS License

<pre>

README.1ST file information

LittleCMS core is released under MIT License

Little CMS

Copyright (c) 1998-2022 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN

NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The below license applies to the following files:
liblcms/cmssm.c

Copyright 2001, softSurfer (www.softsurfer.com)

This code may be freely used and modified for any purpose providing that this copyright notice is included with it. SoftSurfer makes no warranty for this code, and cannot be held liable for any real or imagined damage resulting from its use. Users of this code must verify correctness for their application.

</pre>

AUTHORS File Information
^^^

Main Author

Marti Maria

Contributors

Bob Friesenhahn
Kai-Uwe Behrmann
Stuart Nixon
Jordi Vilar
Richard Hughes
Auke Nauta
Chris Evans (Google)

Lorenzo Ridolfi
Robin Watts (Artifex)
Shawn Pedersen
Andrew Brygin
Samuli
Suominen
Florian Hch
Aurelien Jarno
Claudiu Cebuc
Michael Vhrel (Artifex)
Michal Cihar
Daniel Kaneider
Mateusz Jurczyk (Google)
Paul Miller
Sbastien Lon
Christian Schmitz
XhmikosR
Stanislav Brabec (SuSe)
Leonhard Gruenschloss (Google)
Patrick Noffke
Christopher James Halse Rogers
John Hein
Thomas Weber (Debian)
Mark Allen
Noel Carboni
Sergei Trofimovic
Philipp Knechtges

Special Thanks

Artifex software
AlienSkin software
Jan Morovic
Jos Vernon (WebSupergoo)
Harald Schneider (Maxon)
Christian Albrecht
Dimitrios Anastassakis
Lemke Software
Tim Zaman

...

All images in this directory are copyright 1995 by Jeff Dinkins.
COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

=====

PNG Reference Library License version 2

-
- * Copyright (c) 1995-2022 The PNG Reference Library Authors.
 - * Copyright (c) 2018-2022 Cosmin Truta.
 - * Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
 - * Copyright (c) 1996-1997 Andreas Dilger.
 - * Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted
to use, copy, modify, and distribute
this software, or portions hereof, for any purpose, without fee,
subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard

Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabudde
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are

Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

GIFLIB v5.2.1

GIFLIB License
...

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<https://sourceforge.net/p/giflib/code/ci/master/tree/openbsd-reallocarray.c>

Copyright (c) 2008 Otto Moerbeek <otto@drijf.net>

SPDX-License-Identifier: MIT

CUP Parser Generator for Java v 0.11b

CUP Parser Generator License

...

Copyright 1996-2015 by Scott Hudson, Frank Flannery, C. Scott Ananian, Michael Petter

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

...

This is an open source license. It is also GPL-Compatible (see entry for

"Standard ML of New Jersey"). The portions of CUP output which are hard-coded into the CUP source code are (naturally) covered by this same license, as is the CUP runtime code linked with the generated parser.
^^^

All images in this directory are copyright 1995 by Jeff Dinkins.
Unauthorized reproduction is prohibited.

For more information about Jeff's photographs, please see:

<http://www.theFixx.org/Jeff>

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also,

for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its

terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under

Section 2) in object code or executable form under the terms of Sections 1 and

2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source

code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or

any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In

such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author Gnomovision comes  
with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free  
software, and you are welcome to redistribute it under certain conditions;  
type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here

is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General Public
License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are
subject to the following
clarification and special exception to the GPL, but
only where Oracle has expressly included in the particular source file's header
the words "Oracle designates this particular file as subject to the "Classpath"
exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making
a combined work based on this library. Thus, the terms and conditions of
the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent modules,
and to copy and distribute the resulting executable under terms of your
choice, provided that you also meet, for each linked independent module,
the terms and conditions of the license of that module. An independent
module is a module

which is not derived from or based on this library. If
you modify this library, you may extend this exception to your version of
the library, but you are not obligated to do so. If you do not wish to do
so, delete this exception statement from your version.

Apache Commons Byte Code Engineering Library (BCEL) Version 6.5.0

Apache Commons BCEL Notice

<pre>

Apache Commons BCEL
Copyright 2004-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

</pre>

Apache 2.0 License

<pre>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity.

For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution

notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty

or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

</pre>

Unicode Common Local Data Repository (CLDR) v39

CLDR License

...

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement.
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT.
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE

THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2021 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,

ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

----- Terms of Use -----

-

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the Unicode Privacy Policy.

Unicode Copyright

Copyright 1991-2021 Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

Unicode

Software ("SOFTWARE") includes any source code published in the Unicode Standard or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions

(Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government

is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR

SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

Trademarks & Logos

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

Miscellaneous

Jurisdiction and Venue. This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and

damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

Modification by Unicode, Inc. Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

Taxes. The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

Severability. If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

Entire

Agreement. This Agreement constitutes the entire agreement between the parties.

Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Oracle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ASM Bytecode Manipulation Framework v8.0.1

ASM License

<pre>

Copyright (c) 2000-2011 France Tlcom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY

AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

</pre>

Dynalink v.5

Dynalink License

<pre>

Copyright (c) 2009-2013, Attila Szegedi

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
- * Neither the name of the copyright holder nor the names of
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL COPYRIGHT HOLDER
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

</pre>

Independent JPEG Group: JPEG release 6b

JPEG License

Copyright (C) 1991-1998, Thomas G. Lane.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group.

IJG is not affiliated with the official ISO JPEG standards committee.

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that "The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

International Components for Unicode (ICU4J) v67.1

ICU4J License

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2020 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either

- (a) this copyright and permission notice appear with all copies of the Data Files or Software, or
- (b) this copyright and permission notice appear in associated

Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE

AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.

From: <https://www.unicode.org/copyright.html>:

Unicode Copyright and Terms of Use

For the general privacy policy governing access to this site, see the [Unicode Privacy Policy](#).

Unicode Copyright

Copyright 1991-2020

Unicode, Inc. All rights reserved.

Definitions

Unicode Data Files ("DATA FILES") include all data files under the directories:

<https://www.unicode.org/Public/>

<https://www.unicode.org/reports/>

<https://www.unicode.org/ivd/data/>

Unicode Data Files do not include PDF online code charts under the directory:

<https://www.unicode.org/Public/>

Unicode Software ("SOFTWARE") includes any source code published in the Unicode Standard
or any source code or compiled code under the directories:

<https://www.unicode.org/Public/PROGRAMS/>

<https://www.unicode.org/Public/cldr/>

<http://site.icu-project.org/download/>

Terms of Use

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any
person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming

to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the Unicode DATA FILES and SOFTWARE can be found in the Unicode Data Files and Software License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions (Unicode 5.0 and earlier), these are found on the back of the title page.

The Unicode PDF online code charts carry specific restrictions. Those restrictions are incorporated as the first page of each PDF code chart.

All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

Restricted Rights Legend

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

Warranties and Disclaimers

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode, Inc. may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION E.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE, INC. AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

Waiver of Damages

In no event shall Unicode, Inc. or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode, Inc. was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

Trademarks & Logos

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy)

are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

Miscellaneous

Jurisdiction and Venue. This website is operated from a location in the State of California, United States of America. Unicode, Inc. makes no representation that the materials are appropriate for use in other locations. If you access this website from other locations, you are responsible for compliance with local laws. This Agreement, all use of this website and any claims and damages resulting from use of this website are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this website shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

Modification by Unicode, Inc. Unicode, Inc. shall have the right to modify this Agreement at any time by posting it to this website. The user may not assign any part of this Agreement without Unicode, Inc.'s prior written consent.

Taxes. The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

Severability. If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

Entire Agreement. This Agreement constitutes the entire agreement between the parties.

...

Mesa 3-D Graphics Library v21.0.3

Mesa License

...

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Attention, Contributors

When contributing to the Mesa project you must agree to the licensing terms of the component to which you're contributing.

The following section lists the primary components of the Mesa distribution and their respective licenses.

Mesa Component Licenses

Component	Location	License
Main Mesa code	src/ mesa/	MIT
Device drivers	src/ mesa/ drivers/ *	MIT, generally
Gallium code	src/ gallium/	MIT
Ext headers	GL/ glxext.h	Khronos
	GL/ glxext.h	Khronos
	GL/ wglxext.h	Khronos
	KHR/ khrplatform.h	Khronos

include/GL/gl.h :

Mesa 3-D
graphics library

Copyright (C) 1999-2006 Brian Paul All Rights Reserved.
Copyright (C) 2009 VMware, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR

OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
include/GL/glext.h
include/GL/glxext.h
include/GL/wglxext.h :
```

Copyright (c) 2013 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

```
include/KHR/khrplatform.h :
```

Copyright (c) 2008 - 2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do

so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

Mozilla Public Suffix List

Public Suffix Notice

You are receiving a copy of the Mozilla Public Suffix List in the following
file: <java-home>/lib/security/public_suffix_list.dat. The terms of the
Oracle license do NOT apply to this file; it is licensed under the
Mozilla Public License 2.0, separately from the Oracle programs you receive.
If you do not wish to use the Public Suffix List, you may remove the
<java-home>/lib/security/public_suffix_list.dat file.

The Source Code of this file is available under the
Mozilla Public License, v. 2.0 and is located at
https://raw.githubusercontent.com/publicsuffix/list/3c213aab32b3c014f171b1673d4ce9b5cd72bf1c/public_suffix_list.dat.

If a copy of the MPL was not distributed with this file, you can obtain one
at <https://mozilla.org/MPL/2.0/>.

Software distributed under the License is distributed on an "AS IS" basis,
WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
for the specific language governing rights and limitations
under the License.

MPL v2.0

Mozilla Public License Version 2.0

=====

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to,

deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms

of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3.

Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the

Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must:

(a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become

compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

*

*

* 6. Disclaimer of Warranty

*

* -----

*

*

*

* Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an

*

essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

*

*

*

*

* 7. Limitation of Liability *

* ----- *

* *

* Under no circumstances and under no legal theory, whether tort *

* (including negligence), contract, or otherwise, shall any *

* Contributor, or anyone who distributes Covered Software as *

* permitted above, be liable to You for any direct, indirect, *

* special, incidental, or consequential damages of any *

character *

* including, without limitation, damages for lost profits, loss of *

* goodwill, work stoppage, computer failure or malfunction, or any *

* and all other commercial damages or losses, even if such party *

* shall have been informed of the possibility of such damages. This *

* limitation of liability shall not apply to liability for death or *

* personal injury resulting from such party's negligence to the *

* extent applicable law prohibits such limitation. Some *

* jurisdictions do not allow the exclusion or limitation of *

* incidental or consequential damages, so this exclusion and *

* limitation may not apply to You. *

* *

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License.

Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

...

Copyright (c) %YEARS% Oracle and/or its affiliates. All rights reserved.
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.

This code is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 only, as published by the Free Software Foundation.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License version 2 for more details (a copy is included in the LICENSE file that accompanied this code).

You should have received a copy of the GNU General Public License version 2 along with this work; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA or visit www.oracle.com if you need additional information or have any questions.

Bert Belder: wepoll v 1.5.8

wepoll License

...

wepoll - epoll for Windows
<https://github.com/piscisaureus/wepoll>

Copyright 2012-2020, Bert Belder <bertbelder@gmail.com>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

...

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2021 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,

DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

1.195 kotlin-script-runtime 1.8.10

1.195.1 Available under license :

Apache-2.0

1.196 kotlin-scripting-common 1.8.10

1.196.1 Available under license :

Apache-2.0

1.197 kotlin-scripting-jvm 1.8.10

1.197.1 Available under license :

Apache-2.0

1.198 busybox 1.36.1-r2

1.198.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source of bzip2 version 1.0.4. bzip2 source is distributed under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to

this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.
```

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.199 pgv-java-grpc 0.6.1

1.199.1 Available under license :

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.200 musl-utils 1.2.4-r1

1.200.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

Copyright 2005-2019 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Authors/contributors include:

A. Wilcox
Alex Dowad
Alex Suykov
Alexander Monakov
Andre McCurdy
Andrew Kelley
Anthony G. Basile
Aric Belsito
Arvid Picciani
Bartosz Brachaczek
Benjamin Peterson
Bobby Bingham
Boris Brezillon
Brent Cook
Chris Spiegel
Clment Vasseur
Daniel Micay
Daniel Sabogal
Daurnimator
David Carlier
David Edelsohn
Denys Vlasenko
Dmitry Ivanov
Dmitry V. Levin
Drew DeVault
Emil Renner Berthing

Fangrui Song
Felix Fietkau
Felix Janda
Gianluca Anzolin
Hauke Mehrrens
He X
Hiltjo Posthuma
Isaac Dunham
Jaydeep Patil
Jens Gustedt
Jeremy Huntwork
Jo-Philipp Wich
Joakim Sindholt
John Spencer
Josiah Worcester
Julien Ramseier
Justin Cormack
Kaarle Ritvanen
Khem Raj
Kylie McClain
Leah Neukirchen
Luca Barbato
Luka
Perkov
M Farkas-Dyck (Strake)
Mahesh Bodapati
Markus Wichmann
Masanori Ogino
Michael Clark
Michael Forney
Mikhail Kremnyov
Natanael Copa
Nicholas J. Kain
orc
Pascal Cuoq
Patrick Oppenlander
Petr Hosek
Petr Skocik
Pierre Carrier
Reini Urban
Rich Felker
Richard Pennington
Ryan Fairfax
Samuel Holland
Segev Finer
Shiz
sin
Solar Designer

Stefan Kristiansson
Stefan O'Rear
Szabolcs Nagy
Timo Ters
Trutz Behn
Valentin Ochs
Will Dietz
William Haddon
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy_el.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The implementation of DES for crypt (`src/crypt/crypt_des.c`) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/crypt/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsart.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the elcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

1.201 ssl-client 1.36.1-r2

1.201.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source of bzip2 version 1.0.4. bzip2 source is distributed under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of
Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.202 libssl3 3.1.2-r0

1.202.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users
at the mercy of those companies. By contrast, our General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. The
General Public License applies to the Free Software Foundation's
software and to any other program whose authors commit to using it.
You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not
price. Specifically, the General Public License is designed to make
sure that you have the freedom to give away or sell copies of free
software, that you receive source code or can get it if you want it,
that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender
the rights.

These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain

that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive

use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer

the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY
SERVICING,
REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to humanity, the best way to achieve this is to make it
free software which everyone can redistribute and change under these
terms.

To do so,
attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 1, or (at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free
Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.203 libcrypto3 3.1.2-r0

1.203.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special

exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 1, or (at your option)  
any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of
```

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your

modifications in the Standard Version of the Package.

b) use the modified

Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote

products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.204 openssl 3.1.2-r0

1.204.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it,

that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of

transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
- b) use the modified Package only within your corporation or organization.
- c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
- b) accompany the distribution with the machine-readable source of the Package with your modifications.
- c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output

from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2023 Cisco Systems, Inc. All rights reserved.